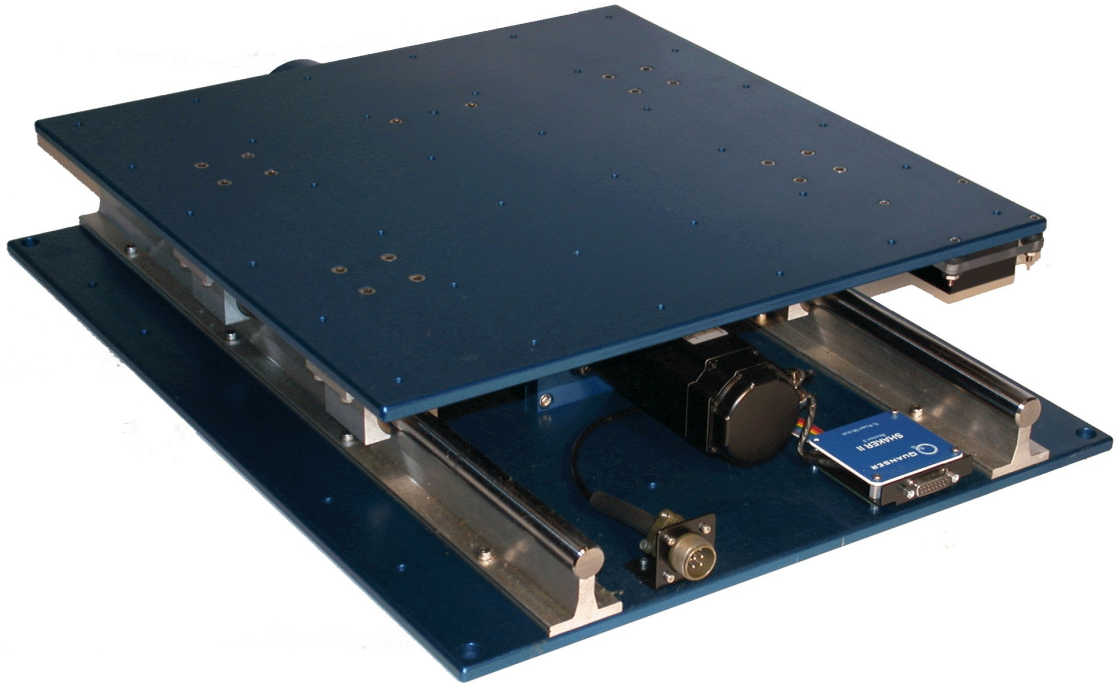
Specialty Plant: Shake Table II

Position Control and Earthquake Analysis

Shake Table II



User Manual

How to contact Quanser:

	+1 (905) 940-3575	Telephone
	+1 (905) 940-3576	Facsimile
	80 Esna Park Drive, # 1-3 Markham, ON Canada L3R 2R6	Mail
	http://www.quanser.com	Web
	mailto://info@quanser.com	General information

Quanser Shake Table II User Manual

MATLAB is a registered trademarks of The MathWorks, Inc.
Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

© 2007 Quanser Inc.
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the copyright holder, except under the terms of the associated software license agreement. No part of this manual may be photocopied or reproduced in any form.
The use of general descriptive names, trade names, trademarks, etc. in this publication, even if the former are not especially identified, is not to be taken as a sign that such names as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Printed in Canada.

Table of Contents

1. Introduction 1

2. Prerequisites 2

3. System Description 3

3.1. Component Nomenclature 3

3.2. Component Description 5

3.2.1. Stage (Component #1) 5

3.2.2. Bottom Plate (Component #2) 5

3.2.3. Brushless DC Motor (Component #3) 5

3.2.4. Lead Screw (Component #4) 6

3.2.5. Encoder 6

3.2.6. Linear Bearing Block (Component #8) 6

3.2.7. Limit Switches (Components #10, 11, and 12) 6

3.2.8. Accelerometer (Component #15) 6

3.3. Specifications 6

4. System Setup 8

4.1. Software Requirements 8

4.1.1. Basic Configuration 8

4.1.2. Advanced Configuration 9

4.2. Hardware Setup 9

4.2.1. Cable Nomenclature 9

4.2.2. Connecting the Q8 Extended Terminal Board 11

4.2.3. Shake Table II Wiring Procedure 12

5. Performing Shake Table II Experiments with WinCon 15

5.1. Overview of Shake Table II CD 15

5.2. Standard ST II Experiments....16
5.2.1. Initializing UPM.... 17
5.2.2. Calibrating Stage.... 18
5.2.3. Sine Wave.... 19
5.2.4. Sine Sweep.... 20
5.2.5. Sample Earthquake.... 22
5.2.6. Working with Scopes.... 23
5.2.7. Optional: Using the GUI.... 27
5.3. ST II Test.... 27

6. Creating Custom ST II Experiments.... 28
6.1. Overview of WinCon and Building a WCL.... 28
6.2. Files Supplied.... 30
6.3. Shake Table II WinCon Library....32
6.3.1. Shake Table II Subsystem.... 33
6.3.2. Accelerometers Subsystem.... 35
6.3.3. Limit Switches Subsystem.... 37
6.3.4. Enable Mode Subsystem.... 39
6.3.5. Calibration Mode Subsystem.... 39
6.3.6. PD Position Control Subsystem.... 40
6.3.7. Acceleration from Encoder Subsystem.... 42
6.3.8. Scope Subsystems.... 43
6.3.9. Stop after Duration Subsystem.... 44
6.4. Matlab Scripts....44
6.4.1. Setup Script: setup.m.... 45
6.4.2. Generating Composite Sine Wave: make_sine.m.... 52
6.4.3. Constructing Scaled Earthquake: make_quake.m.... 54
6.4.4. FFT Evaluation: fft_eval_pos.m and fft_eval_acc.m.... 61
6.5. Sine Wave: q_sine_zz.mdl....64
6.5.1. Model Description.... 64
6.5.2. Building and Running Sine Wave WinCon Controller.... 67
6.6. Sine Sweep: q_sweep_zz.mdl....69

6.6.1. Model Description..... 69
6.6.2. Building and Running Sine Sweep WinCon Controller..... 70
6.7. Predefined Data: q_data_zz.mdl..... 71
6.7.1. Model Description..... 71
6.7.2. Building and Running Predefined Trajectory WinCon Controller..... 75
6.7.3. Constructing a WinCon Project..... 77

7. References..... 78

Appendix A. Troubleshooting Guide..... 79

Appendix B. Q8 Extended Terminal Board Signals..... 80

Illustration Index

Figure 1: Shake Table II system....1
Figure 2: Overview of major system components....2
Figure 3: Top view of Shake Table II components....4
Figure 4: Front view of Shake Table II components....4
Figure 5: "Motor" cable....10
Figure 6: "To Device" cable....10
Figure 7: "From MultiQ" cable....10
Figure 8: "Emergency Stop" cable....10
Figure 9: "From Analog Sensors" cable....11
Figure 10: Connections on the Q8 Extended Terminal Board....13
Figure 11: Connections on UPM-180-25B....13
Figure 12: Connections of the Shake Table II device....14
Figure 13: Message prompted after running UPM boot WCP....17
Figure 14: Messaged prompted after running calibration WCP....18
Figure 15: Front panel of sine wave WCP....19
Figure 16: The x (m) scope when running the q_sine_q8 controller....20
Figure 17: Control panel for sine sweep project....21
Figure 18: The a_tbl (g) scope after running the q_sweep_q8 controller....22
Figure 19: Acceleration measurements plotted in a_tbl (g) scope after running Northridge earthquake....23
Figure 20: Open plot button in the WinCon Server window....25
Figure 21: Selecting scopes to be opened....26
Figure 22: Changing scope time buffer window....26
Figure 23: GUI for browsing through ST II WinCon controllers....27
Figure 24: Building a WinCon Project from a Simulink Diagram....28
Figure 25: Simulink model used to track sine wave....30
Figure 26: Shake Table II WinCon library....33
Figure 27: Shake Table II - Q8 subsystem....34
Figure 28: Software watchdog: stops WinCon when encoder reads over P_MAX variable....35
Figure 29: Accelerometer subsystem in WinCon Shake Table II Library....36
Figure 30: Subsystem used to remove initial bias and filter accelerometer signal....36
Figure 31: Limit switches subsystem in WinCon ST II Library....37
Figure 32: Demonstrating the digital debounce switch....38
Figure 33: Limit Switches wo/ Safety subsystem in WinCon ST II library....39
Figure 34: Proportional-derivative controller subsystem in ST II WinCon Library....40
Figure 35: Block diagram of the control system used to position of ST II stage....41
Figure 36: Calculate acceleration from position measurement subsystem in ST II WinCon Library....42
Figure 37: Positions scopes subsystem in WinCon ST II Library....43
Figure 38: Acceleration scopes in WinCon ST II Library....43
Figure 39: Stop controller after duration subsystem in WinCon ST II Library....44
Figure 40: Dependencies and various M-File function calls....45

Figure 41: For a 0 kg load, this plot illustrates the maximum setpoint amplitude over a range of frequencies..... 48
Figure 42: Composite sine wave plot generated by the make_sine.m script.....54
Figure 43: Steps to generating the setpoint trajectory from a raw earthquake file.....55
Figure 44: Plot generated by q_scale.p showing generated acceleration and scaled position.....57
Figure 45: Browsing for an earthquake on the Berkeley website.....58
Figure 46: Searching for an earthquake on the Berkeley website..... 59
Figure 47: Web page showing data collected for Kobe tremor at the HIK station..... 60
Figure 48: Saving earthquake acceleration record..... 60
Figure 49: Raw earthquake data file HIK000.AT2 shown when opened in Matlab Editor..... 61
Figure 50: Bode plot of desired and measured stage position after running q_data_q8 with a predefined sine wave.....63
Figure 51: Bode plot of the Northridge earthquake acceleration data.....64
Figure 52: The q_sine_q8 Simulink Model used to control the stage position according to a user-specified sine wave..... 65
Figure 53: Shake Table II - Control System - Q8 subsystem.....66
Figure 54: Smooth Sine Setpoint subsystem in q_sine_q8 Simulink Model..... 67
Figure 55: Compiling model message.....68
Figure 56: WinCon Server window when loaded with a controller and ready to run.....68
Figure 57: The q_sweep_q8 Simulink model used for sine sweep position control on the ST II.....69
Figure 58: Sine Sweep Setpoint subsystem in q_sweep_q8 Simulink Model..... 70
Figure 59: The q_data_q8 Simulink model used for predefined trajectory position control on the ST II... 72
Figure 60: From Workspace Setpoint block in q_data_zz Simulink Diagram..... 73
Figure 61: Shake Table II - Control System with FF - Q8 subsystem in q_data_q8 Simulink Diagram..... 74
Figure 62: Block diagram of proportional-derivative plus feed-forward position ST II control system..... 75
Figure 63: Desktop view after running a sine wave with the q_data_q8 controller.....76

Index of Tables

Table 1: Component nomenclature..... 3
Table 2: Dimensions of the top stage..... 5
Table 3: Dimensions of the bottom plate..... 5
Table 4: Shake Table II specifications..... 7
Table 5: Cable nomenclature..... 11
Table 6: Ribbon cable connections between Quanser Q4/Q8 HIL board and Q8 Extended Terminal Board..... 12
Table 7: Shake Table II wiring summary..... 15
Table 8: Description of main folder on the Shake Table II CD..... 15
Table 9: Files supplied in the "wcp" folder of the Shake Table II CD..... 17
Table 10: Description of the predefined scopes in the ST II WinCon Projects..... 24
Table 11: Summary of file types..... 29
Table 12: List of files provided in the "mdl" folder on the Shake Table II CD..... 32
Table 13: Sample controls gains calculated for a varying load mass..... 49
Table 14: Equations used to find the maximum setpoint amplitude..... 52
Table 15: Signals in the Table X connection..... 80
Table 16: Signals in Table Y connection..... 81

1. Introduction

The Quanser Shake Table II shown in Figure 1 is an instructional shake table device that was originally developed for the *University Consortium on Instructional Shake Tables* (UCIST). It can be used to teach structural dynamics, vibration isolation, feedback control, and various other topics for mechanical, aerospace, and civil engineers.

The top stage of the shake table is driven by a powerful motor that allow it to achieve an acceleration of 2.5 g when up to 7.5 kg of mass is mounted. The stage rides on two ground-hardened metal shafts using linear bearings which allows for smooth linear motions with low path deflection. When starting from center the stage is capable of moving 7.62 cm, or 3-inches, on each side. It therefore has a total travel of 15.24 cm. In order to move the top platform at a high acceleration, a robust ball-screw and motor assembly is used. The high-powered 400 Watt motor is a 3-phase brushless DC actuator. The motor contains an embedded high-resolution encoder that allows the position of the stage of be measured with an effective linear resolution is 3.10 μm. An analog accelerometer is mounted on the Shake Table II platform in order to measure the acceleration of the stage directly.

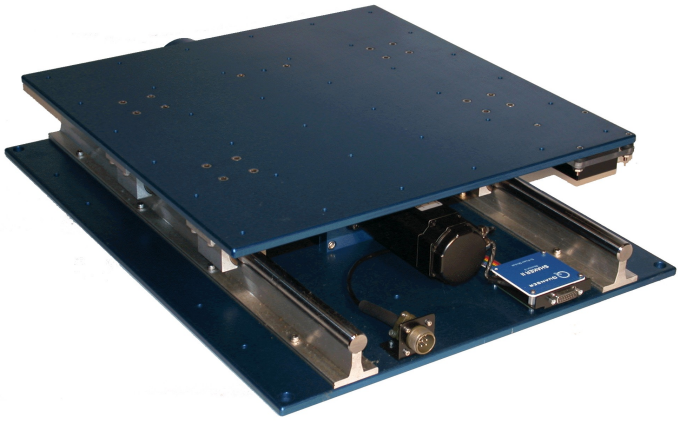


Figure 1: Shake Table II system.

The main devices used to run the shake table is depicted in Figure 2. The entire is comprised of a Universal Power Module (UPM), a data-acquisition (DAC) card, a PC running the WinCon control software, and the Shake Table II itself. Consider for example the signal transitions between the system components when the user wishes the Shake Table II stage to track a sine wave and read the resulting acceleration. Using WinCon on the PC, the user specifies the amplitude and frequency of the sine wave. The sine wave is the user-specified desired or command position of the stage. The current needed to move the stage at the desired sine wave position is calculated in WinCon and sent through the analog output channel of the data-acquisition board to the UPM device. The power amplifier in the UPM amplifies the current and drives the motor. The table moves back and forth at the position and frequency of the commanded sine wave. The resulting displacement and acceleration of the stage are

measured by the on-board encoder sensor and the accelerometer sensor. The encoder and accelerometer are connected to the DAC board and their signals can be displayed and processed further in WinCon. Plotted data can be also be saved for later analysis.

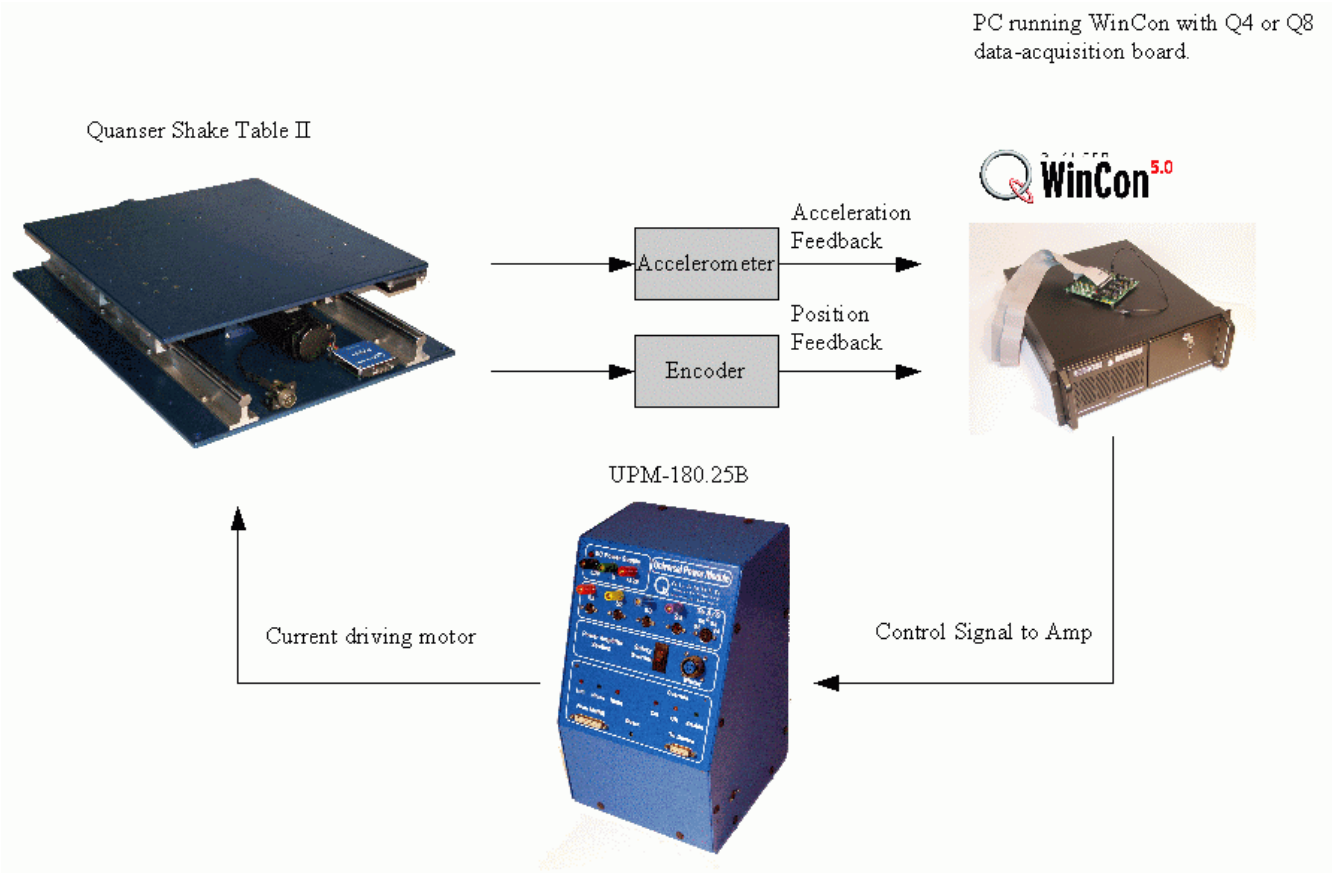


Figure 2: Overview of major system components.

Section 3 describes the various devices composing the Shake Table II system and gives some specifications. The wiring procedure for a typical setup is explained in Section 4. In Section 5, the procedures to run the basic experiments on the shake table are described. It explains, for instance, how how to make the top stage of the shake table track a sine wave. Section 6 goes into more detail and explains how to design custom shake table controllers. For example, the user may wish to have the shake table track a sawtooth wave instead of a sine wave. This is not already supplied and would have to be created. In addition, Section 6 discusses how to download recorded earthquake data from the Internet and replay them on the shake table.

2. Prerequisites

To successfully run a Shake Table II controller, the prerequisites are:

i. To be familiar with your Shake Table II main components (e.g. actuator, sensors), your data acquisition card (e.g. Q8), and your power amplifier (e.g. UPM), as described in this manual, References [1], and [2], respectively.
ii. To be familiar with WinCon enough to open a WinCon Project and run it to control the position of the shake table device. This is explained in this manual.
iii. For the *advanced* configuration (to be explained later), users need to be familiar in using WinCon to control and monitor the plant in real-time and in designing their controller through Simulink. Reference [3] provides full details of the WinCon software.

3. System Description

The main components that compose the Shake Table II device are identified in Section 3.1 and those components are described in Section 3.2. See Section 3.3 for a listing of the Shake Table II specifications.

3.1. Component Nomenclature

Table 1 provides a list of all the principal elements composing the Shake Table II system. The components are identified on the Shake Table II system in figures 3 and 4 through the corresponding identification (ID) numbers.

ID #	*Description*	*ID #*	*Description*
1	Stage	**9**	Sensor Circuit Board
2	Base Plate	**10**	Left Limit Sensor
3	Brushless DC Motor	**11**	Home Position Sensor
4	Lead screw	**12**	Right Limit Sensor
5	Ball nut	**13**	Motor Leads Connector
6	Manual adjustment.	**14**	Motor Encoder & Hall Sensors Connector
7	Hardened Steel Guide Rail	**15**	Accelerometer
8	Linear bearing block	**16**	Accelerometer Connector

Table 1: Component nomenclature.

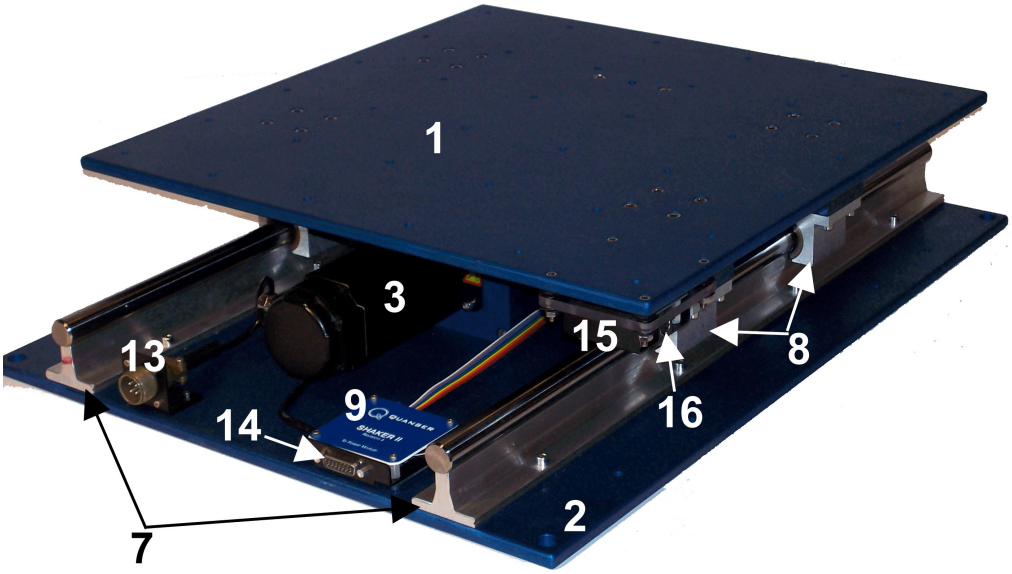


Figure 3: Top view of Shake Table II components.

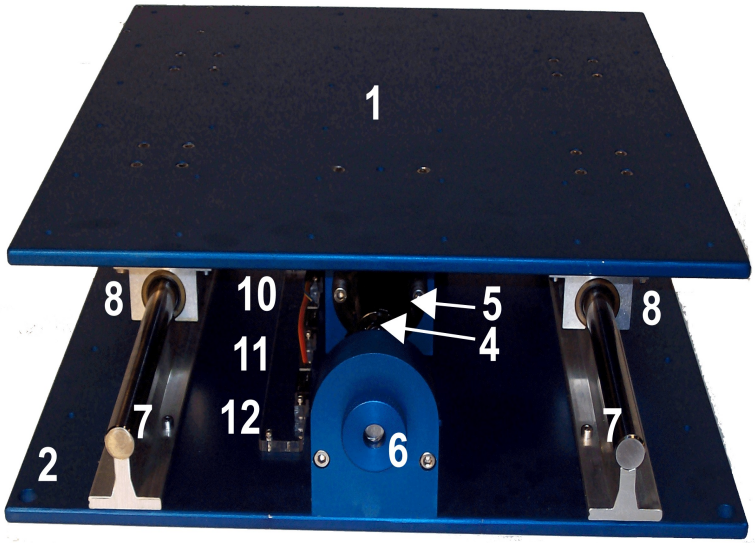


Figure 4: Front view of Shake Table II components.

3.2. Component Description

This section describes some of the components listed in Table 1 above.

3.2.1. Stage (Component #1)

Table 2 below lists and characterizes the overall dimensions of the 18-inch by 18-inch top stage or platform of the Shake Table II system. It is labeled with ID #1 and shown in figures 3 and 4. As illustrated, the top plate has many screw holes that can be used to fasten structures and other objects such as the Quanser AMD-1 and AMD-2 structures (see Reference [5] and [6]).

Description	*Value*	*Unit*
Length	0.457	m
Width	0.457	m
Thickness	0.0097	m

Table 2: Dimensions of the top stage.

3.2.2. Bottom Plate (Component #2)

The bottom support plate of the Shake Table II is pictured in figures 3 and 4 by ID #2 and its dimensions are given in Table 3. The metal shafts as well as the ball-screw are all fastened onto this plate. Notice that there are four large screw holes at each corner as well as smaller screw holes along the sides of the plate. These can be used to fasten the shake table onto a ground floor support to prevent the shake table system from moving, or at least reduce the amount of vibration. Although this is not necessary, it is recommended in order to yield more precise results when, for instance, measuring acceleration.

Description	*Value*	*Unit*
Length	0.609	m
Width	0.457	m
Thickness	0.00124	m

Table 3: Dimensions of the bottom plate.

3.2.3. Brushless DC Motor (Component #3)

The Shake Table II incorporates a Kollmorgen AKM24 brushless 3-phase DC Motor, as shown in Figure 3 by ID #3. The motor has a power of 400 Watts. It is connected to a ball-screw that and drives the ball nut assembly fastened to the bottom of the shake table platform. The brushless commutation is ensured through three hall sensors with a phasing of 120 degrees. Some of the motor specifications are included in Table 4. More detailed motor specifications are available in Reference [4].

3.2.4. Lead Screw (Component #4)

The lead screw, shown by ID #4 in Figure 4, circulates through a ball nut, component #5 in Figure 4, that is attached to the bottom of the shake table stage and is rotated by the motor. The lead screw has a pitch of 0.50 inches. Thus the shake table stage moves 0.50 inches, or 1.24 cm, per single ball-screw revolution.

3.2.5. Encoder

The encoder is embedded inside the DC motor and is used to measure the linear position of the stage. It cannot be identified externally. The motor encoder has a resolution of 2048 lines per revolution. In quadrature mode this gives 8192 counts per full rotation motor shaft rotation. The effective resolution, i.e. minimum linear position that can be detected, of the stage displacement is 3.10 μm.

3.2.6. Linear Bearing Block (Component #8)

For smooth motion, the stage is fitted with low friction linear ball bearing blocks, shown by ID #8 in figures 4 and 5, that glide on two ground hardened shafts, identified by component #7 in figures 3 and 4.

3.2.7. Limit Switches (Components #10, 11, and 12)

These *Left*, *Home*, and *Right*, proximity sensors are shown in Figure 4 with ID # 10, 11, and 12, respectively. The *Left* or *Right* limit switch get triggered when the top stage moves close to the left or right mechanical range. Similarly, the *Home* sensor is set to high when the top stage is at the mid-stroke or center position. These sensors are used by UPM-180-25B to deactivate the power amplifier for safety as well as for calibrating the stage to its center position.

3.2.8. Accelerometer (Component #15)

An accelerometer is mounted underneath the top stage of the Shake Table II that measures the acceleration in both the *x* and *y* directions. It is shown with ID #15 label in Figure 3. The analog sensor is calibrated such that 1 Volt equals 1 *g*, or 9.81 m/s^2.

CAUTION: The accelerometer readings can be misleading and lead to unexpected results. Please use caution when using them as they are generally used as indicators.

3.3. Specifications

The main specifications of the Shake Table II are given in Table 4. See Reference [2] for UPM-180-25B specifications and Reference [4] for a complete listing of the DC motor characteristics.

Parameter	*Matlab Notation*	*Parameter Description*	*SI Value*	*Units*	*IM Value*	*Units*
R_m	Rm	Motor armature resistance.	2.94	ohm		
K_t	Kt	Motor current-torque constant.	0.360	N.m/A	3.2	lb.in/A
K_m	Km	Motor back-emf constant.	0.2034	V/(rad/s)	23.4	V/k_{rpm}
P_b	Pb	Ball-screw pitch.	0.0127	m/rev	0.5	in/rev
M_p	Mp	Preload mass.	7.74	kg	17.1	lb
M_{l_max}	Ml_max	Maximum total load mass.	15.0	kg	33.0	lb
M_s	Ms	Mass of Shake Table II system.	27.2	kg	60.0	lb
		Dimension of top stage.	0.46 ×0.46	m^2	18.0 ×18.0	in^2
		Dimension of bottom stage.	0.61 ×0.46	m^2	24.0 ×18.0	in^2
		Height from bottom to top stage.	12.4	cm	4.875	in
x_{max}	P_MAX	Maximum stroke position.	76.2	mm	3.0	in
v_{max}	VEL_MAX	Maximum linear velocity of stage.	664.9	mm/s	26.18	in/s
F_{max}	F_MAX	Maximum linear force of stage.	708.7	N	159.3	lb
a_{max}	ACC_MAX	Maximum linear acceleration of stage for 0 kg load.	24.5	m/s^2		
g_{max}	G_MAX	Maximum linear acceleration of stage for 0 kg load.	2.50	g		
K_{ENC}	K_ENC	Encoder sensitivity gain	3.1006	µm/count	1.22E-004	in/count
K_{ACC}	K_ACC	Accelerometer sensitivity gain	-1	g/V		
		Dynamic load capacity of ball nut.	12000	N	2697.6	lb
		Life expectancy of ball nut at full load.	6.35E+008	m	2.50E+010	in
		Life expectancy of linear bearing.	6.35E+006	m	2.50E+008	in
		Load carrying capacity of linear bearings.	131.5	kg	290	lb

Table 4: Shake Table II specifications.

4. System Setup

To setup this experiment, the following hardware and software are required:

- ✔ **Power Amplifier:** Quanser UPM 180-25B, as detailed in Reference [2].
- ✔ **Data Acquisition Board:** Quanser Q4 or Q8 board. See Reference [1] for details.
- ✔ **Shake Table II Plant:** Quanser Shake Table II, as shown in Figure 1 above.
- ✔ **Real-Time Control Software:** WinCon-RTX configuration detailed in Reference [3].

Follow these steps to setup the Shake Table II system:

1. Install the Quanser Q4/Q8 board in a PC as described in Reference [1]. Reference [1] also provides a detailed description of the Q4/Q8 board.

CAUTION: In many cases a PC is shipped with the STII system that is already setup with WinCon and either a Q4 or Q8 board. In that case, the Q4/Q8 board does not need to be installed..

2. Install the WinCon control software on the PC and its associated programs as described in Reference [10]. Also see Section 4.1 for a listing of the necessary software needed depending on whether the *basic* or *advanced* configuration is to be setup.

CAUTION: In many cases a PC is shipped with the STII system that is already setup with WinCon. In that case, WinCon does not need to be installed.

3. Connect the Quanser UPM-180-25B and the Shake Table II to the Q4/Q8 data-acquisition system as explained in Section 4.2. The Quanser UPM-180-25B is the power amplifier used to drive the shake table motor and is described in Reference [2]

4.1. Software Requirements

There are two system configurations: **basic** and **advanced**. The *basic* configuration only enables users to run previously made WinCon controllers. It does not let users create or modify new ones as in the *advanced* configuration. Further, the *advanced* configuration enables users to perform more advanced analysis such as finding the FFT of a signal.

4.1.1. Basic Configuration

As already mentioned, the *basic* setup allows users to run the previously made WinCon controllers that are supplied with the Shake Table II CD. For example, users can load the *q_sine.wcp* file with WinCon to make the table track a sine wave. In this system configuration however, the functionality of the existing controllers cannot be modified and new controllers cannot be created. Thus a new controller that makes the table track a sawtooth wave command could not be constructed.

Software components for the *basic* configuration:

- ✔ Windows 2000/XP
- ✔ RTX
- ✔ WinCon

See Reference [10] for details on the compatible Windows and RTX versions needed to run WinCon.

4.1.2. Advanced Configuration

In the *advanced* setup, users can run the previously constructed WinCon controllers. In addition, users gain the ability to modify existing and create new WinCon controllers. The Matlab/Simulink package along with Visual Studio .NET is required for this. Here is the necessary software components for the *advanced* configuration:

- ✔ Windows 2000/XP
- ✔ RTX
- ✔ WinCon
- ✔ Matlab loaded with the following:
    - ✔ Simulink
    - ✔ Real-time Workshop (RTW)
- ✔ Visual Studio .NET

See Reference [10] for details on the compatible Windows, RTX, Matlab/Simulink, and Visual Studio .NET versions needed to run WinCon.

4.2. Hardware Setup

The different cables used to connect the various components of the shake table system is described in Section 4.2.1. Section 4.2.2 explains how to connect the Quanser Q8 Extended Terminal Board to either the Quanser Q4 or Q8 data acquisition card. In Section 4.2.3, the typical connections between the Q4 or Q8 board, the UPM-180-25B device, and the Shake Table II are described.

4.2.1. Cable Nomenclature

The various connector cables that are provided with the Shake Table II system are listed in Table 5. If additional system were ordered, such as the AMD-1 or AMD-2 (see references [5] and [6]), then other cables not listed Table 5 will have been supplied.

Cable	*Description*
 Figure 5: "Motor" cable.	The "Motor" cable corresponds to the 3-phase motor power leads. This cable is designed to connect from the Quanser's Universal Power Module model 180-25B (i.e. the output of the power module after signal amplification) to the brushless DC motor of the shake table.
 Figure 6: "To Device" cable	The "To Device" cable is a DB15 cable that connects the shake table circuit board to the UPM. It carries to the UPM the three limit sensors' signals and the motor encoder signals. It also supplies the DC power required by the different sensors.
 Figure 7: "From MultiQ" cable.	The "From MultiQ" 25-pin parallel cable connects the UPM to the data acquisition card terminal board. It is compatible with Quanser's quick-connect system. It carries the motor encoder signals, limit sensor signals, calibrate signal, and the S1, S2, S3, and S4 analog signals from the UPM. From the DAC, the cable carries the control signal (to be amplified and sent to the motor) as well as the calibrate and enable digital signals.
 Figure 8: "Emergency Stop" cable.	The "Emergency Stop" cable has a 6-pin-mini-DIN connector that connects to the side of the UPM. The UPM is enabled when the safety pushbutton switch is not pressed.

Cable	Description
Figure 9: "From Analog Sensors" cable.	The "From Analog Sensors" cable is a 6-pin-mini-DIN-to-6-pin-mini-DIN cable that can be used to connect any potential plant sensor to the UPM such as accelerometers. It can provide a ±12VDC bias to analog sensors and carry their voltage signals to the DAC terminal board via the UPM.

Table 5: Cable nomenclature.

4.2.2. Connecting the Q8 Extended Terminal Board

Before wiring the ST II system, the Q4 or Q8 data-acquisition card must first be installed and connected to Q8 Extended Terminal Board. This terminal board is made specifically to interface with the Quanser UPM-180-25B device and is different then the standard terminal boards used.

If not already done so, see Reference [1] for instructions on how to install the Q4 or Q8 hardware-in-the-loop board in your PC. Then, follow these instructions to connect the Q4 or Q8 hardware-in-the-loop board to the Q8 Extended Terminal Board:

1. Ensure that the PC is powered off and that you are grounded.
2. Connect the ribbon cable labeled **J3** into the **Cable 3** connector on the Extended Terminal Board. Cable 3 is the shortest cable. In a tower PC, it will be the topmost cable where it comes out of the PC.

CAUTION: The red line on the ribbon cable should be closest to the "Table X" and "Table Y" parallel input connectors on the terminal board. The connectors have a tab such that they only go in one way.

3. Table 6 summarizes the connections between the Q4 and Q8 HIL board and the Q8 Extended Terminal Board. If using a Q8 control board, connect the **J2** ribbon cable from the Q8 into the **Cable 2** connector on the Q8 Extended Terminal Board. Cable 2 is the second shortest cable. In a tower unit, it is the middle cable where the ribbon cables exit the PC. If using a Q4 control board, there is no J2 cable and the Cable 2 connector on the terminal board is left unused.
4. Connect the **J1** ribbon cable to the **Cable 1** connector on the Q8 Extended Terminal Board. This is the longest ribbon cable.
5. With the terminal board sitting on top of the tower unit, the ribbon cables should not be twisted and should all lie neatly on top of one another. The cables are slightly offset from one another on the terminal board to match the offset of the connectors on the data acquisition card. The text on the terminal board should face the front of the computer.
6. Use the ribbon cable clamp provided to secure the ribbon cables together. The clamp comes with a peel-and-stick base so that you can mount the clamp to a clean surface, such as the top of your PC. Put the cables in the clamp before mounting it. If you do not wish to mount the clamp, then do not remove the peel-and-stick paper. Use of the clamp is unnecessary, but helps to keep your Q8 data acquisition system neat and tidy.

As mentioned in the procedure, in the Q4 DAQ board setup there is no **J2** ribbon cable. As a result, the **Cable 2** connector on Q8 Extended Terminal Board is not used.

Quanser HIL Board	*Connections to Q8 Extended Terminal Board*
Q8	J1 -> CHANNEL 1 J2 -> CHANNEL 2 J3 -> CHANNEL 3
Q4	J1 -> CHANNEL 1 J3 -> CHANNEL 3

Table 6: Ribbon cable connections between Quanser Q4/Q8 HIL board and Q8 Extended Terminal Board.

See Reference [1] for instruction on how to install the necessary drivers for the Q4 or Q8 board. Once the Q4/Q8 has been properly tested, you can proceed to Section 4.2.3 and wire the ST II system.

4.2.3. Shake Table II Wiring Procedure

The procedure in Section 4.2.3.1 explains in detail how to connect the Shake Table II system to run experiments. Section 4.2.3.2 provides a brief summary of the required connections.

4.2.3.1. Detailed Wiring Procedure

This section describes in detail how to connect the Shake Table II, UPM-180-25B and Q8 Extended Terminal Board together. The connections are illustrated in figures 10, 11, and 12 with a corresponding identification number. Follow these steps to wire the ST II system:

1. **Connect cable #1** from the "Table X" connector on the Quanser Q8 Extended Terminal Board, shown in Figure 10, to the "From MultiQ" connector on the Quanser UPM-180-25B, as depicted Figure 11. This connection is done using the parallel cable shown in Figure 7. See Appendix B for a listing of all the signals that are passed through this cable.
2. **Connect cable #2** from the "To Device" connector located on the UPM front panel, as shown in Figure 11, to the ST II circuit board illustrated in Figure 12. This cable used in this connection is the DB15 cable described in Table 5 and pictured in Figure 6. It carries all three proximity sensor signals, the motor encoder signals, and the brushless motor hall sensor signals to the UPM-180-25B unit. The motor encoder is used to calculate the linear position of the cart and is denoted by the variable x.
3. **Connect cable #3** into the side of the UPM, as presented in Figure 11. Cable #3 is the "Emergency Stop" cable described in Table 5 and illustrated in Figure 8. The UPM is active if and only if the remote E-Stop switch is *depressed*.
4. **Connect cable #4** from the "Motor" connector located on the UPM front panel, as depicted in Figure 11, to the "Motor" connector on the shake table, as shown Figure 12. The motor leads connector is component #13 in Figure 3. This connection is done using the "Motor" cable described in Table 5 and illustrated in Figure 5. It carries the required 3-phase power to the

brushless motor.

5. **Connect cable #5** from the analog connector on the accelerometer mounted on the shake table , see Figure 12, to the "S1" connector on the front panel of the UPM-180-25B. **Ensure the UPM is not powered when making this connection.** It carries the acceleration measured by the accelerometer.
6. Go through the ST II test procedure in Section 5.3.

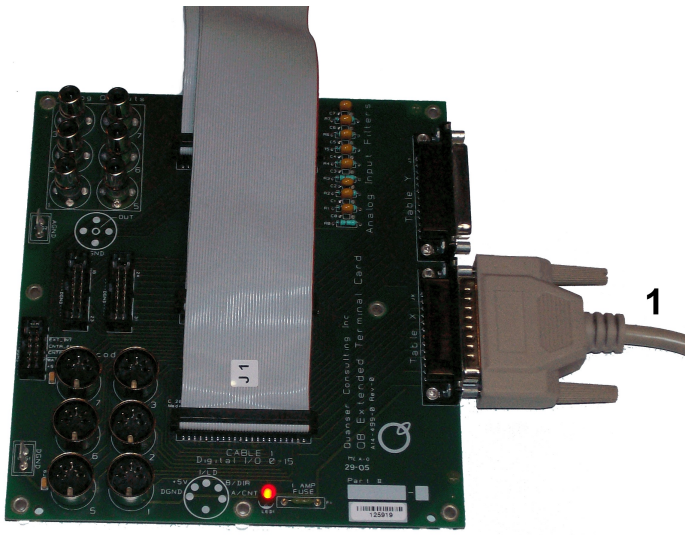


Figure 10: Connections on the Q8 Extended Terminal Board.

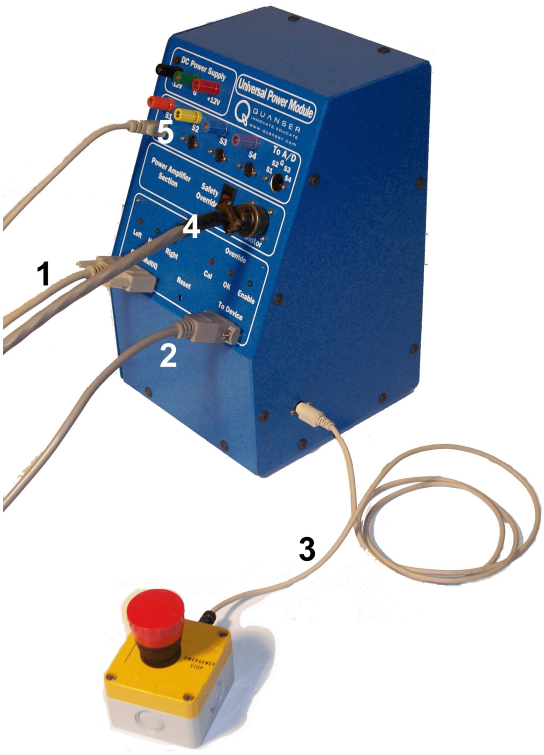


Figure 11: Connections on UPM-180-25B.

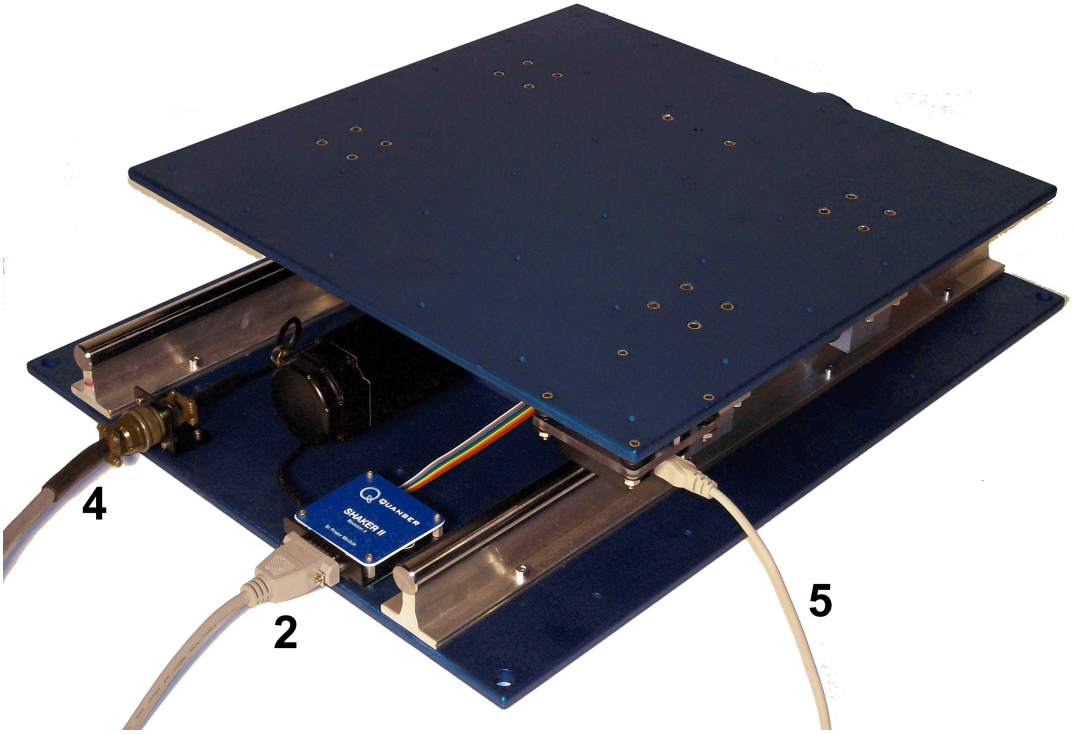


Figure 12: Connections of the Shake Table II device.

4.2.3.2. Wiring Summary

See Table 7 below for a summary of the Shake Table II connections.

Cable #	*Cable Type*	*From*	*To*	*Function*
1	25-pin Parallel Cable	"Table X" on the Terminal Board	"From MultiQ" on the blue UPM-180-25B	Drives the amplifier to move the stage and receives the accelerometer, stage encoder, calibration, and limit detector signals from UPM.
2	15-pin Parallel Cable	"To Device" on the blue UPM-180-25B	Circuit board on the Shake Table II	Receives the encoder and limit detector signals from the shake table.
3	"Emergency Stop" Cable	E-Stop Switch	UPM E-Stop Connector	Carries the emergency stop signal.
4	4-pin Motor Cable	"Motor" Connector on the	Motor connector on the Shake	Connects the shake table motor leads to the amplifier on the UPM.

Cable #	Cable Type	From	To	Function
		UPM-180-25B	Table II	
5	Analog Cable: 6-pin-mini-DIN to 6-pin-mini-DIN	"S1" Connector on the on the UPM-180-25B	Accelerometer on the Shake Table II	Carries the acceleration signal of the stage to the UPM.

Table 7: Shake Table II wiring summary.

5. Performing Shake Table II Experiments with WinCon

Section 5.1 briefly describes the directory structure of the Shake Table II CD. All users, both those with the *basic* and *advanced* system configurations, should go through the standard experiments listed in Section 5.2 and the procedure in Section 5.3 to test the Shake Table II connections.

5.1. Overview of Shake Table II CD

The *Shake Table II CD* supplied with the system contains various files. There are three folders on the shake table CD that correspond to three different system configurations and they are explained in Table 8.

Folder Name	Description
STII	Contains this manual as well as controllers to run the Shake Table II system by itself.
AMD	Has manuals and controllers for using the Quanser Active Mass Damper by itself. The active mass damper, i.e. The linear cart on top of the building, perturbs the building and then dampens for the building oscillations. The one floor structure is called AMD-1 and is described in Reference [5] and the two floor structure is called the AMD-2 device and is presented in Reference [6].
STII+AMD	Contains manuals and controllers to use the Quanser AMD-1 and AMD-2 systems when mounted on top of the Shake Table II system. See Reference [7] and [8] for more information on the STII+AMD-1 and STII+AMD-2 systems. In this configuration, the shake table provides the base motions to perturb the building while the active mass does the vibration dampening.

Table 8: Description of main folder on the Shake Table II CD.

The *STII* folder on the *Shake Table II CD* has the following directory structure:

\STII

..... *Manual*
..... *UPM Manual*
..... *Specifications*
..... *Lab Files*
..... *q4*
.... *wcp*
.... *mdl*
..... *q8*
.... *wcp*
.... *mdl*

The Manual folder contains documentation such as this *Shake Table II User Manual*, the *Universal Power Module User Manual* manual, and the motor specification sheet. The *Lab Files* directory includes previously compiled WinCon controllers that were built for the Q4 and Q8 boards, in the *wcp* folder, as well as the source Simulink models that were used to create those WinCon controllers, in the *mdl* folder. The *wcp* name stands for WinCon Project and the *mdl* folder name denotes Simulink Model. Choose the lab files depending on the whether the Quanser Q4 or Q8 HIL board is connected to the ST II.

5.2. Standard ST II Experiments

There are five standard controllers for the Shake Table II: initializing the UPM, calibrating the stage to the home position, running a sine wave, running a sine sweep, and running an earthquake. These standard experiments are contained in the *wcp* folder of the Shake Table II CD. The WinCon Project file name for each experiment is listed in Table 9 below. It also includes a description of the experiment (what WinCon Project does when ran) and the section number where the procedure to run the WinCon Project is given. Only the *basic* configuration is required to run these experiments (Matlab is not needed).

File Name	*Description*	*Section Number*
q_boot_upm_zz.wcp	Initializes the UPM-180-25B to make the amplifier ready-to-be-enabled. This has to be done prior to performing any of the ST II experiments.	5.2.1
q_cal_zz.wcp	Returns the stage to the *home* position. The stage should be at the home position before running any of the experiments.	5.2.2
q_sine_zz.wcp	Position of the stage tracks a sine wave with an amplitude and frequency set by the user.	5.2.5
q_sweep_zz.wcp	Sends a sine sweep to the shake table for generating the frequency response.	5.2.4

File Name	Description	Section Number
q_quake_kobe_zz.wcp	Kobe earthquake.	5.2.5
q_quake_northridge_zz.wcp	Northridge earthquake.	5.2.5
start_exp.m	Opens a graphical user interface that can be used to open the various WinCon Projects. **Matlab is needed to run this**.	5.2.7

Table 9: Files supplied in the "wcp" folder of the Shake Table II CD.

In addition, see Section 5.2.6 for more information on using scopes. It lists a description of all the scopes available for viewing, explains how to open new scopes, and how to adjust the time axis of a plot.

5.2.1. Initializing UPM

When the blue UPM-180-25B is first powered, the *Left* and *Right* LEDs located on its front panel should be blinking. To stop the blinking and initialize the UPM-180-25B device, run the *q_boot_upm_zz.wcp* project as instructed in the Reference [2] in Section 3.5.1. The steps are summarized here:

1. Ensure that the *Safety Override* switch, located on the UPM front panel, is OFF.
2. Connect the *Emergency Stop* cable to the connector on the side panel of the UPM.
3. Rotate the knob in the counter-clockwise direction until it is released in the upright position. The amplifier cannot drive the motor when the red knob is pushed in.
4. After power up, the *Left* and *Right* LEDs on the UPM front panel should be blinking. If the lights are NOT flashing consult Reference [2].
5. Load the WinCon Server software (typically placed under Quanser | WinCon in the Windows START menu).
6. Open the *q_boot_upm_zz.wcp* WinCon project that is supplied. The *zz* suffix denotes the type of data-acquisition card being used. For example, if *zz* is *q4* then the file is meant to be used with the Quanser Q4 control board. Alternatively, if *zz* is *q8* then the file is to be used with the Quanser Q8 data-acquisition system.
7. Run the initialization controller by clicking on the green START button in the WinCon Server window.
8. The *Left* and *Right* LEDs should stop flashing and the window shown in Figure 13 should be prompted. If the LEDs are no longer blinking, then the UPM amplifier is ready to be used for the various ST II experiments.

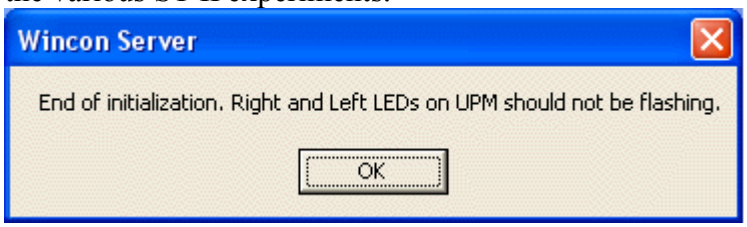


Figure 13: Message prompted after running UPM boot WCP.

5.2.2. Calibrating Stage

Before running any of the experiments the stage of the Shake Table II should be in the mid-stroke position. This position is called the *Home* position because the Home limit sensor is triggered when the stage is centered.

Follow this procedure to calibrate the stage to the *Home* position:

1. Ensure the UPM180-25B has been initialized as instructed in Section 5.2.1.
2. Load the WinCon Server software (usually under Quanser | WinCon in the Windows START menu).
3. In the WinCon Server window, click on File | Open and select the WinCon Project file *q_cal_zz.wcp*.
4. Click on the green *START* button in the *WinCon Server* window.

ATTENTION: PRESS DOWN on the RED BUTTON of the E-Stop switch in case of emergency. If something goes wrong during an experiment, pressing the red button of the E-Stop switch disables the amplifier and shuts off the DC motor power.

5. On the front panel of the UPM, the LEDs *Cal*, *OK*, and *Enable* LEDs should all be lit and the stage should begin moving. If the Left or Right limit sensor was already triggered, then the stage begins to immediately move towards the center. If no limit switch was initially triggered, then the stage will begins to move towards Left limit sensor. Once the Left limit sensor is hit, the stage reverses its direction and begins moving towards the mid-stroke position. The stage stops moving when the *Home* limit switch is triggered (the *Home* LED on the UPM will go ON). When complete, the message shown in Figure 14 is displayed.

Figure 14: Messaged prompted after running calibration WCP.

6. Alternatively, the knurled knob at the end of the table can be used to manually return the table to its home position. The *Home* LED on the UPM will be lit when this position is reached.
7. If the table is not moving consult the troubleshooting guide at the end of this manual.

When the UPM is placed in the calibration mode, i.e. the *Cal*, *OK*, and *Enable* LEDs are lit, the amplifier remains enabled when the *Left* or *Right* proximity sensor are triggered but is disabled when the *Home* limit sensor is activated. Normally and for all the other experiments, the amplifier is disabled when the *Left* or *Right* sensor is triggered.

5.2.3. Sine Wave

In this experiment, the position of the Shake Table II stage tracks a user-specified sine wave signal. The user can specify the amplitude and frequency of the sine wave.

Follow these steps to run the sine wave project:

1. Make sure the amplifier has been initialized as discussed in Section 5.2.1.

2. **CAUTION: Ensure table is at *HOME* position before running this experiment!** Otherwise the experiment may stop prematurely because the table reached the left or right limit sensors. If the *Home* LED on the front panel of the blue UPM-180-25B is **not** lit, then go through the calibration procedure in Section 5.2.2.
3. Load the WinCon Server software (usually under Quanser | WinCon in the Windows START menu).
4. In the WinCon Server window, click on File | Open and select the WinCon Project file *q_sine_zz.wcp*. The project contains the control panel shown in Figure 15 and the *x (m)* scope, depicted in Figure 16 with results. The control panel has a vertical slider to change the amplitude of the sine wave position command and a knob to control the frequency of the signal.

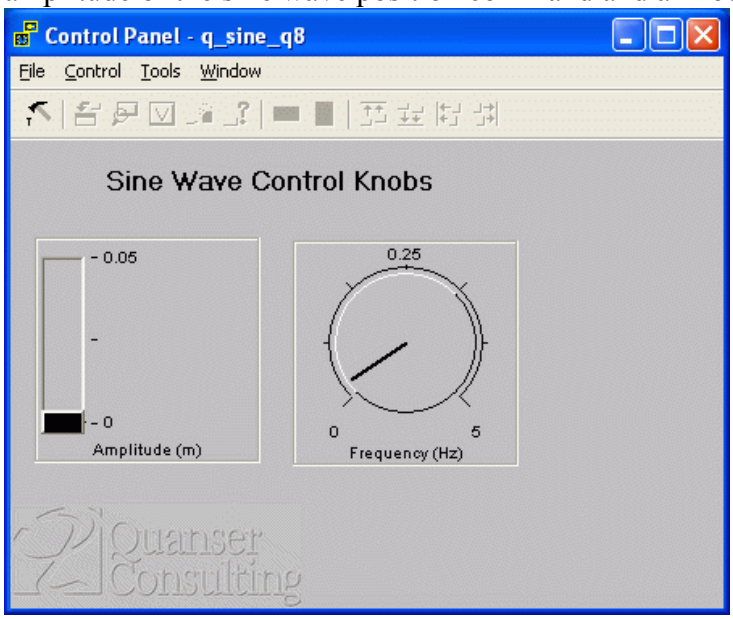


Figure 15: Front panel of sine wave WCP.

5. Click on the green *START* button in the *WinCon Server* window. The *x (m)* scope should begin plotting data and the START button in the WinCon Server window should now be a red STOP button. Since the amplitude in the control panel is set to 0, the table is not commanded any position and therefore should not be moving.

ATTENTION: PRESS DOWN on the RED BUTTON of the E-Stop switch in case of emergency. If something goes wrong during an experiment, pressing the red button of the E-Stop switch disables the amplifier and shuts off the DC motor power.

6. The *Enable* and *OK* LEDs on the UPM should be lit. Consult the troubleshooting guide if this is not the case.
7. Move the vertical slider in the Control Panel to 0.02 meters. The stage should begin tracking a sine wave of 20 cm at a frequency of 0.5 Hz. The *x (m)* scope displays the desired or commanded position set by the user and the resulting measured position. Figure 16 illustrates the results when the sine wave frequency is changed from 0.5 Hz to 1.0 Hz at around the 2 second mark. As shown, both the desired and measured responses are almost identical (however, there is always a small discrepancy due to the sampling delay).

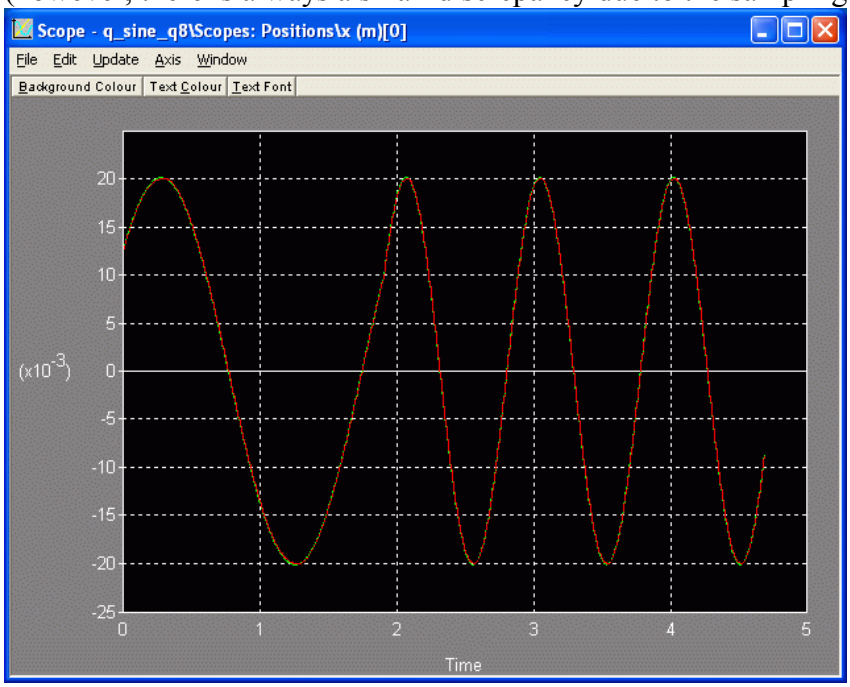


Figure 16: The x (m) scope when running the q_sine_q8 controller.

8. When finished set the *Amplitude (m)* control knob to 0 m, in order to bring the stage back to *Home* position.
9. Click on the red *STOP* button in the *WinCon Server* window to stop running the sine wave controller.
10. Shut off the UPM-180-25B power.

5.2.4. Sine Sweep

The sine sweep, also known as a chirp signal, is a sine wave with a fixed amplitude that increases in frequency as time progresses. In this experiment, the stage of the Shake Table II tracks a given sine sweep that increases from 1 Hz to 15 Hz in 30 seconds. By default the sine amplitude is 0.2 cm but this can be varied though the control panel. Typically the sine sweep is used to find the frequency response of a structure that is mounted on the table stage (need the *advanced* package for this).

Follow this procedure to run the sine sweep controller:

1. Ensure the amplifier has been initialized as discussed in Section 5.2.1.

2. **ATTENTION: Ensure table is at *HOME* position before running this experiment!** Otherwise the experiment may stop prematurely because the table reached the left or right limit sensors. If the *Home* LED on the front panel of the blue UPM-180-25B is **not** lit, then go through the calibration procedure in Section 5.2.2.
3. Load the WinCon Server software (usually under Quanser | WinCon in the Windows START menu).
4. In the WinCon Server window, click on File | Open and select the WinCon Project file *q_sweep_zz.wcp*. The project contains the control panel shown in Figure 17 and the scope a_tbl (g) pictured in Figure 18 with data. With the vertical slider in the control panel, the user can change the amplitude of the sine sweep signal between 0 and 3 cm. The *a_tbl (g)* scope displays the acceleration measured by the table accelerometer in gravitational units, *g*.

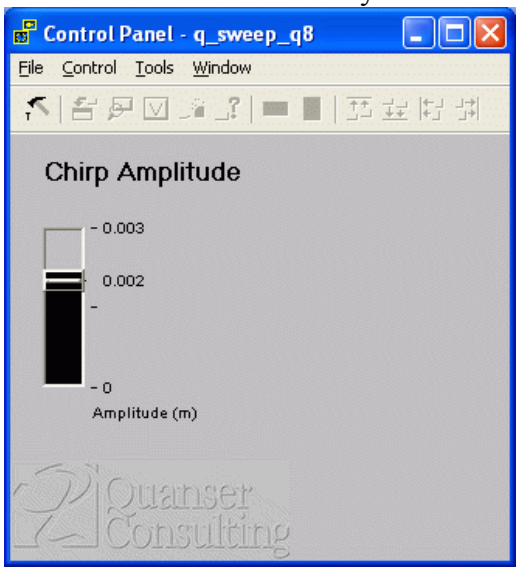


Figure 17: Control panel for sine sweep project.

5. Click on the green *START* button in the *WinCon Server* window to start the controller.

ATTENTION: PRESS DOWN on the RED BUTTON of the E-Stop switch in case of emergency. If something goes wrong during an experiment, pressing the red button of the E-Stop switch disables the amplifier and shuts off the DC motor power.

At first, the stage moves back and forth slowly as it tracks the sine wave and the *a_tbl (g)* scope begins plotting acceleration data. The stage will begin oscillating faster and faster as the frequency of the position command increases over time. The sine sweep lasts 30 seconds and is automatically restarted once the duration is reached. See Figure 18 for a typical acceleration measurement after running the sine sweep on the shake table.

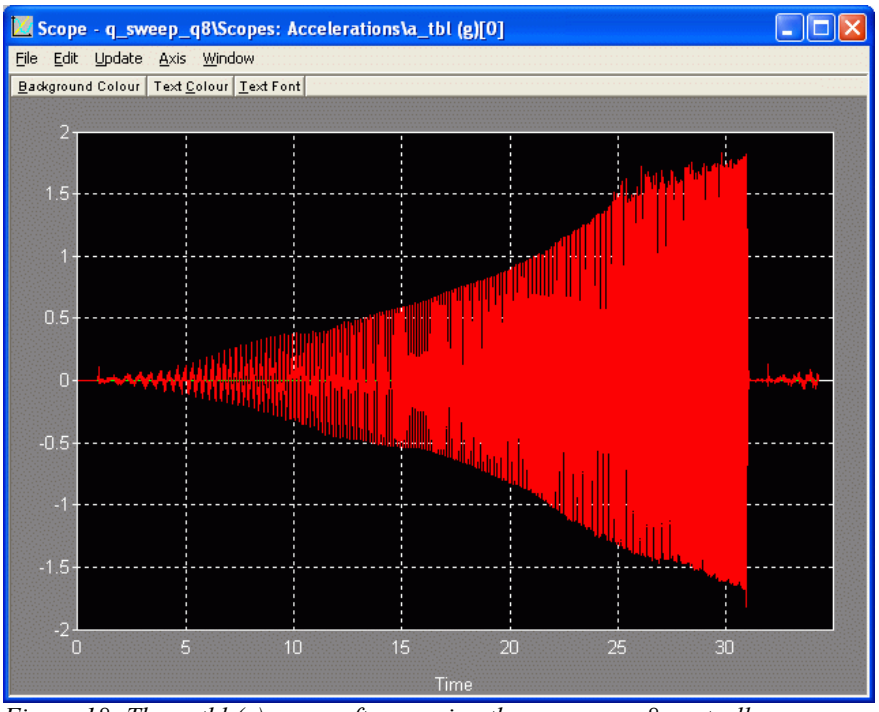


Figure 18: The a_tbl (g) scope after running the q_sweep_q8 controller.

6. Click on the red *STOP* button in the *WinCon Server* window to stop running the sine sweep controller.
7. Shut off the UPM-180-25B power.

5.2.5. Sample Earthquake

Recorded earthquake data that was collected when actual earthquake occurred can be scaled down and ran on the shake table. Two standard historical earthquakes have been supplied for the user: Kobe and Northridge.

Follow this procedure to run a sample earthquake:

1. Ensure the amplifier has been initialized as discussed in Section 5.2.1.

2. **ATTENTION: Ensure table is at *HOME* position before running this experiment!** Otherwise the experiment may stop prematurely because the table reached the left or right limit sensors. If the *Home* LED on the front panel of the blue UPM-180-25B is **not** lit, then go through the calibration procedure in Section 5.2.2.
3. Load the WinCon Server software (usually under Quanser | WinCon in the Windows START menu).
4. In the WinCon Server window, click on File | Open and select either the *q_kobe_zz.wcp* or *q_northridge_zz.wcp* WinCon Project files to run the Kobe or Northridge earthquakes on the table. The projects include the *a_tbl (g)* scope, depicted in Figure 19 after running the experiment, that displays the desired acceleration, which is the actual recorded acceleration of the tremor, in green and the acceleration measured by the table accelerometer in red. Both accelerations are given in gravitational units, *g*, and should be similar.

5. Click on the green *START* button in the *WinCon Server* window.

PRESS DOWN on the RED BUTTON of the E-Stop switch in case of emergency. If something goes wrong during an experiment, pressing the red button of the E-Stop switch disables the amplifier and shuts off the DC motor power.

The stage should begin moving back and forth and the accelerations in the *a_tbl (g)* scope should start plotting. The end result is shown in Figure 19.

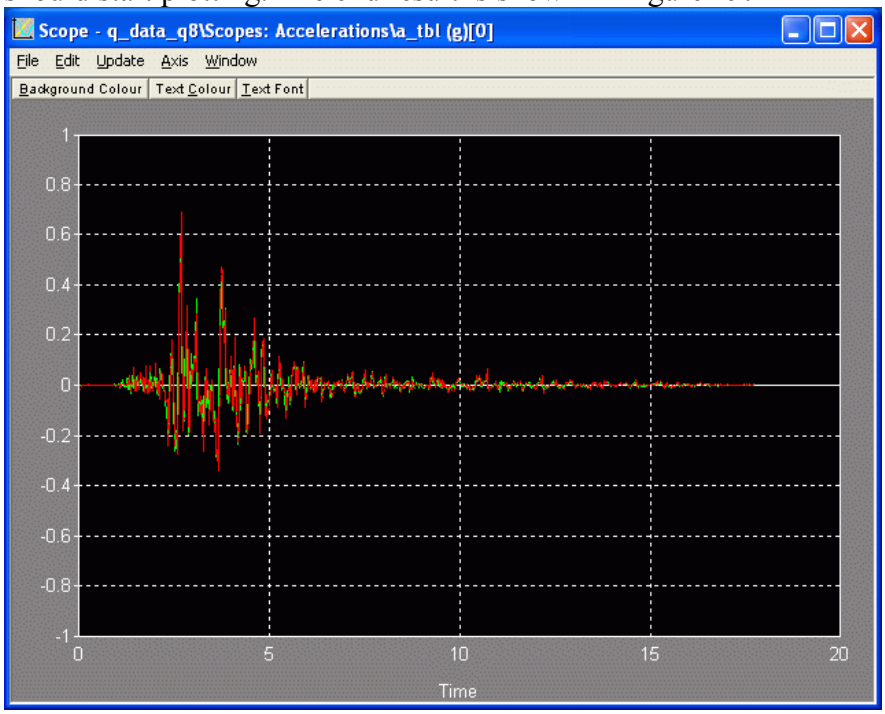


Figure 19: Acceleration measurements plotted in a_tbl (g) scope after running Northridge earthquake.

6. WinCon stops automatically when the earthquake duration has been reached.
7. Shut off the UPM-180-25B power

5.2.6. Working with Scopes

Section 5.2.6.1 describes all the scopes available for viewing and Section 5.2.6.2 explains how to open a new scope (that was perhaps not already opened when loading the WinCon Project). The procedure to decrease or increase the time scale of a scope is described in Section 5.2.6.3.

5.2.6.1. Scope Descriptions

With the exception of the *q_boot_upm_zz*, all the supplied ST II controllers include the scopes listed in Table 10.

Matlab Name	*Variable Name*	*Description*
x (m)	x	Displays two plots: (1) green plot is the desired stage position in meters (2) red plot is the measured stage position in meters
x (in)	x	Same as above except position shown in inches.
V (m/s)	v_x	Measured stage velocity in meters per second calculated from position measurement.
V (in/s)	v_x	Same as above except in inches per second.
x_ddot (g)	$a_{x,enc}$	Acceleration of stage calculated from the position measurement given in gravitational units, *g*.
x_ddot (m/s^2)	$a_{x,enc}$	Same as above except acceleration shown in meters per second squared, m/s^2.
Im (A)	I_m	Input motor current in amperes.
a_tbl (g)	a_g	Displays two plots: (1) green plot is the stage acceleration measured by the accelerometer (2) red plot is the desired acceleration used (set to 0 when using *q_sine_zz* or *q_sweep_zz* controllers).
a_tbl (m/s^2)	a_g	Same as above except acceleration shown in meters per second squared, m/s^2.
a_f1 (g)	a_{f1}	Acceleration of the stage measured by accelerometer #1 in gravitational units, *g*. It is denoted *a_f1* because, typically, it is used with the accelerometer mounted on the first floor of the Quanser AMD-1 or AMD-2 structure (see Reference [5] and [6] for more information).
a_f1 (m/s^2)	a_{f1}	Same as above except acceleration shown in meters per second squared, m/s^2.
a_f2 (g)	a_{f2}	Acceleration of the stage measured by accelerometer #2 in gravitational units, *g*. It is denoted *a_f2* because, typically, it is used with the accelerometer mounted on the second floor of a Quanser AMD-2 structure (see Reference [6] for more information).
a_f2 (m/s^2)	a_{f2}	Same as above except acceleration shown in meters per second squared, m/s^2.

Table 10: Description of the predefined scopes in the ST II WinCon Projects.

As described in Table 10, the *x (m)*, *x (in)*, *a_tbl (g)*, and *a_tbl (m/s2)* scopes display two variables of data. Most of the time, the desired and measured positions in the *x* scopes closely match one another and, as a result, the green (desired) and red (measured) plot traces will be almost directly on top of each

other.

The desired acceleration is only used in controllers with a previously defined trajectory, such as the *q_kobe_zz* or *q_northridge_zz* controllers. When running those experiments, the green desired acceleration trace and the red measured acceleration traces in the *a_tbl (g)*, and *a_tbl (m/s2)* scopes will closely match. However, when running the *q_sine_zz* or *q_sweep_zz* controllers the green desired acceleration scope is set to 0 because there is no desired acceleration defined (only the red measured signal will be shown).

5.2.6.2. Opening Additional Scopes

When opening a WinCon Project such as *q_sine_zz.wcp*, only the *x (m)* scope is loaded. If the user wishes to view, for instance, the acceleration measured by the stage accelerometer in *g* units, then the *a_tbl (g)* scopes has to be opened.

Follow these steps to open one of the scopes defined in Table 10

1. Load one of the supplied WinCon projects (except q_boot_upm or q_cal_zz) as described by the above procedures.
2. Click on the *Open plot* button in the WinCon Server window, as shown in Figure 20 below.

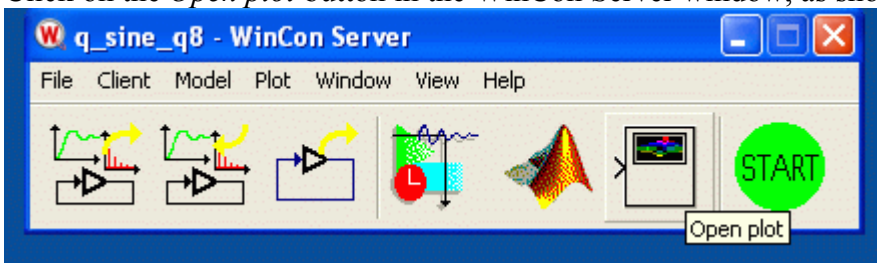


Figure 20: Open plot button in the WinCon Server window.

3. Select the variables to view. For example as illustrated in Figure 21, to view the velocity and acceleration of the stage place a check mark on the *v (m)* and *a_tbl (g)* items.

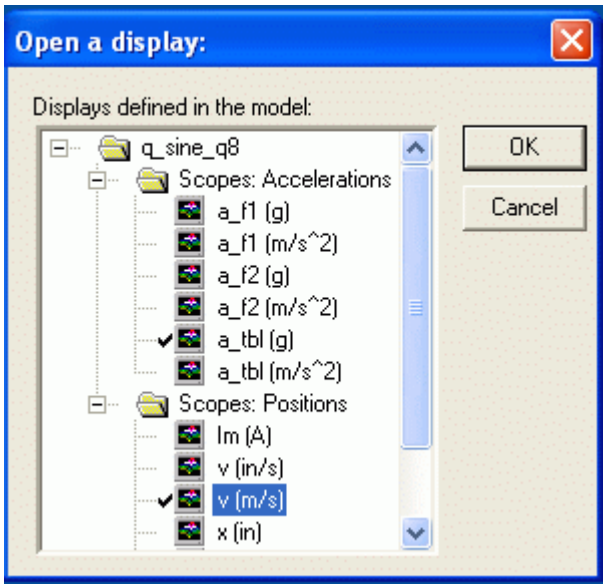


Figure 21: Selecting scopes to be opened.

4. Click on the *OK* button to load the scope(s).

5.2.6.3. Changing Time Scale of a Scope

By default, scopes have a time-scale of 5.0 seconds. The horizontal time-scale, or buffer size, of a scope can be changed in order to view data over a larger span of time or to "zoom up" on a response by decreasing the time-scale.

Follow this procedure to change the time buffer of an opened scope:

1. Select the *Update* | *Buffer...* item in the scope menu bar to load the *Select Buffer Size* window shown in Figure 22.

Figure 22: Changing scope time buffer window.

2. Enter a new time buffer. For example, if 10 seconds is entered instead of 5 then the scope will display 10 seconds of data instead of 5 seconds.
3. Click on the *OK* button to apply the changes.

5.2.7. Optional: Using the GUI

For users with the *Advanced* package, or at least those that have Matlab , the graphical user-interface shown in Figure 23 can be used to browse through the various supplied WinCon Projects.

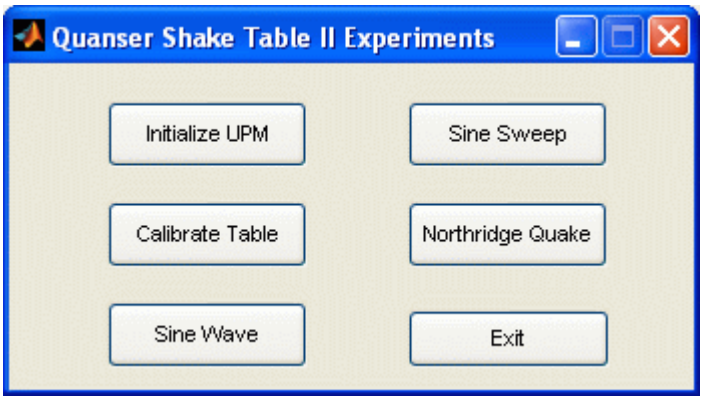


Figure 23: GUI for browsing through ST II WinCon controllers.

Follow this procedure to open the GUI:

1. Load Matlab
2. Through the *Current Directory* window, go to the *STII\Lab Files\zz\wcp* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).
3. Run the *start_exp.m* Matlab script. The GUI shown in Figure 23 should load.
4. Click on a button to load the corresponding WinCon Project. For example, click on the *Sine Wave* button to load the sine controller that is explained in Section 5.2.3.
5. Click on the *EXIT* button to close the GUI.

5.3. ST II Test

This section goes through a procedure to test the Shake Table II hardware after it has been setup as dictated in Section 4. This method should be undergone before performing any of the standards experiments

Follow these steps to test the Shake Table II:

1. Verify that the system has been setup as instructed in Section 4.
2. Turn the blue UPM power ON. The Left and Right LEDs should be blinking. To stop the blinking and initialize the UPM-180-25B device, run the *q_boot_upm_zz* project as instructed in the Section 5.2.1. If the *Left* and *Right* LEDs stop blinking the UPM has been initialized. This also implies that WinCon has been properly installed and that the UPM and Q4/Q8 connection has been made properly. Move on to the next step if this test passed.
3. Run the *q_cal_zz* controller as detailed in Section 5.2.2 to center the stage. If this test works then the Shake Table II and UPM connections have been made properly and the user can continue to the next step.
4. Run the *q_sine_zz* controller explained in Section 5.2.3. If the table tracks the commanded sine wave then the system has been properly installed and is functioning properly.

6. Creating Custom ST II Experiments

In Section 6.1, a short introduction to WinCon is given and the basic procedure to build a WinCon Controller from a Simulink model is described. An overview of the files used to construct WinCon controllers that are supplied with the *Shake Table II* CD is given in Section 6.2. Section 6.3 describes the blocks inside the *WinCon Shake Table II Library* that are used to construct the various Simulink diagram supplied. In Section 6.4, the Matlab scripts that setup the parameters in the Simulink diagrams and so on are described. The q_sine_zz, q_sweep_zz, and q_data_zz Simulink models are explained and the procedure to build and run a WinCon Controller from these files is all given in sections 6.5, 6.6, and 6.7.

CAUTION: Before attempting to build and run your own custom controller, make sure the ST II has been tested as dictated in Section 5.3.

6.1. Overview of WinCon and Building a WCL

WinCon is a software designed by Quanser that can run Simulink models in real-time on a PC. As demonstrated in Figure 24, from a user-designed Simulink diagram WinCon generates real-time code and creates a corresponding WinCon Controller file. The WinCon Controller is a file with extension *.wcl that can be opened and ran using the WinCon Client or WinCon Server software.

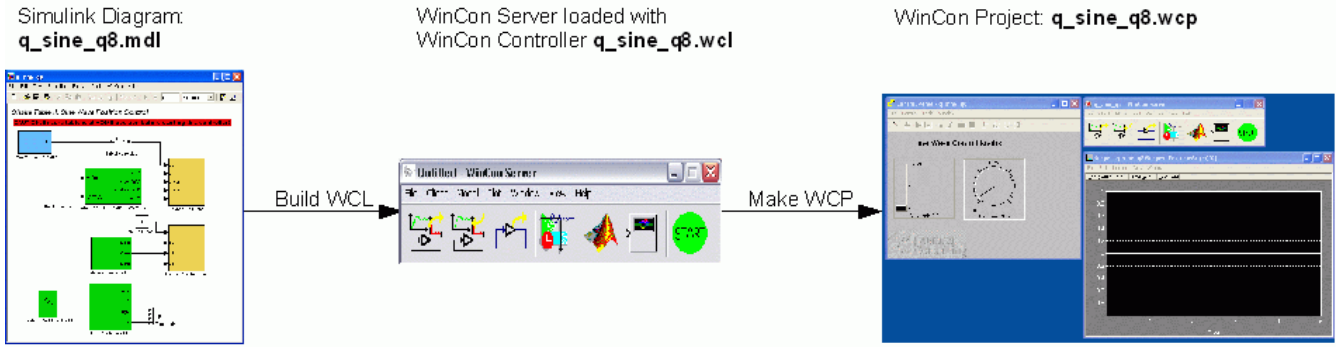


Figure 24: Building a WinCon Project from a Simulink Diagram.

After loading a WinCon controller in WinCon Server, various display sinks such as scopes and digital meters can be opened and created to display data. Control panels can also be created to change the values of the blocks in the Simulink model. Further, these sinks and control panels can be saved under a WinCon Project, which is a file with the extension *WCP*. When the WinCon Project is loaded, any previously defined displays and panels are opened. Figure 24 shows the WinCon Project for q_sine_q8. It has a control panel that controls the amplitude and frequency of the sine wave and a scope that shows the desired and measured position of the Shake Table II stage.

Matlab script files, or M-files, are used to load parameters used by the Simulink diagrams. Matlab preparsed pseudocode files, or P-code, can be generated from Matlab scripts. These files can be run faster than an ordinary M-File but cannot be edited. Table 5 summarizes the various file extensions and describes the associated file.

File Extension	*File Type*	*Description*
mdl	Simulink Diagram	Simulink model of the controller.
m	Matlab script	Used to set gains and various other parameters in the Simulink diagram.
p	Matlab preparsed pseudocode	This is generated from an m-file. It runs faster then a Matlab script but cannot be edited.
wcl	WinCon Controller	Generated from the Simulink model and executed by a WinCon Client software.
wcp	WinCon Project	Contains predefined plots and control panels.

Table 11: Summary of file types.

The *q_sine_q8.mdl* file is the Simulink model shown in Figure 25. It is designed to interact with the Shake Table II system using input and output blocks from the WinCon Toolbox. The stage of the shake table tracks the sine wave commanded through the model. This is done using a control system that outputs a current to the shake table motor such that the position of the stage tracks the desired sine wave position set in the Simulink diagram. The procedure to create the *q_sine_q8* WinCon Controller is:

1. Run the *setup.m* Matlab script to load the various parameters in the Matlab workspace that are used in the Simulink model.
2. As illustrated in Figure 25, the real-time code and the corresponding WinCon Controller is built by clicking on the WinCon menu in the Simulink diagram and selecting the *Build* item.

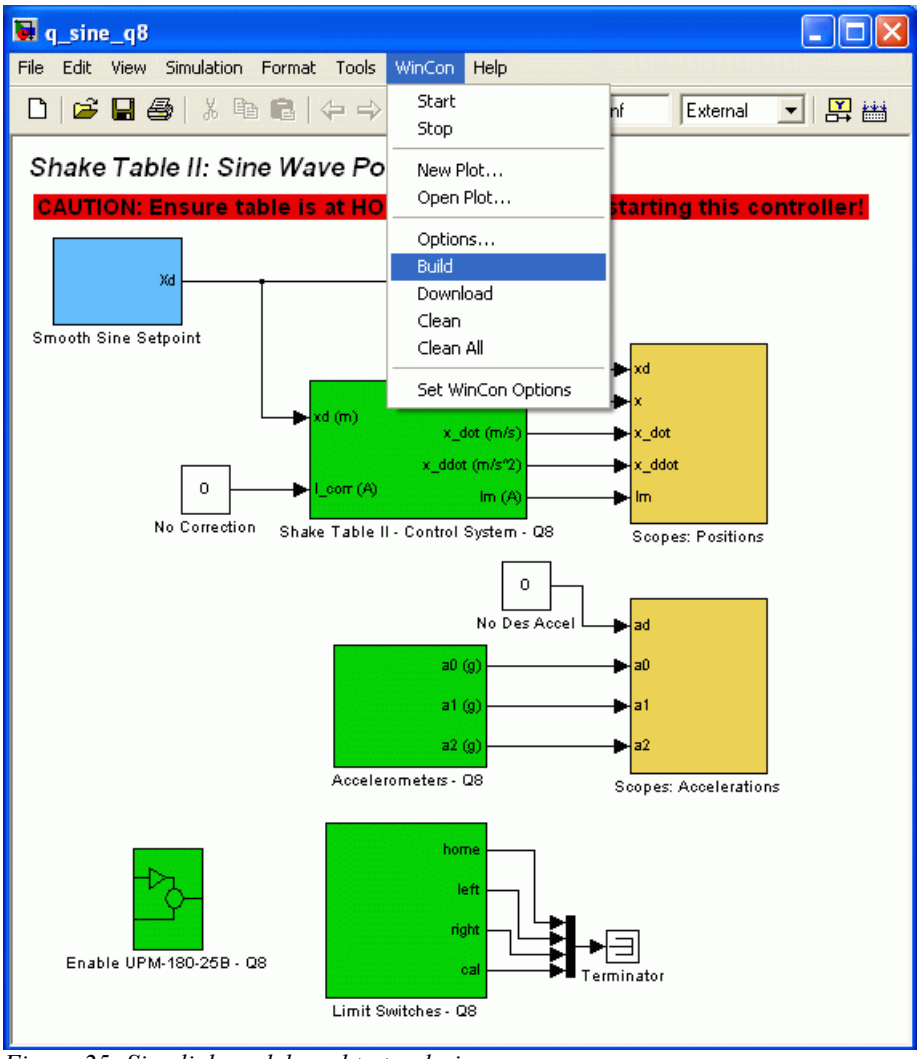


Figure 25: Simulink model used to track sine wave.

3. After the code is finished generating, the WinCon Server window loads and the user can click on a green *START* button to run the digital controller.

This is basically how a controller is implemented on the shake table and gives an overview on the topics covered later. The procedure to building a WinCon Controller will be defined in greater detail later.

6.2. Files Supplied

As summarized in Table 11, there are various types of files involved with running a digital controller to control hardware. The files supplied in the *mdl* folder on the *Shake Table II* CD are listed in Table 12 below. It also includes a short description of the purpose of each file. Note that these files can only be used by users with the *advanced* ST II system configuration.

File Name	*Type*	*Description*
q_boot_upm_zz	Simulink Model	Initializes the UPM-180-25B.
q_cal_zz	Simulink Model	Calibrates stage to home/center position.
q_sine_zz	Simulink Model	Manual sine wave tracking.
q_sweep_zz	Simulink Model	Sine sweep tracking.
q_data_zz	Simulink Model	Predefined data tracking.
calc_conversion_constants	Matlab M-File	Loads useful conversions factors.
compute_control_gains	Matlab M-File	Calculates control gains based on specifications given.
construct_quake_trajectory	Matlab M-File	Constructs a time-based array from a matrix of data and the sampling time.
construct_sine_trajectory	Matlab M-File	Constructs a composite sine wave time-based array given the frequency, amplitude, duration, and sampling time.
dderiv	Matlab M-File	Takes double derivative of input using the least squares technique.
display_results.m	Matlab M-File	Displays encoder resolution, various load mass information, and limits of the shake table.
encoder_calibration.m	Matlab M-File	Calculates encoder sensitivity gain.
fft_eval_acc	Matlab M-File	Creates FFT plot of the desired and measured table accelerations.
fft_eval_pos	Matlab M-File	Creates FFT plot of the desired and measured table positions.
init_earthquake_data	Matlab M-File	Re-formats raw earthquake data file.
load_accelerations	Matlab M-File	Computes maximum acceleration of the load given the amplitude and frequency of a sine wave.
lpf_cutoff_freq	Matlab M-File	Outputs the cutoff frequency and damping ratio used for various low-pass filters.
make_quake	Matlab M-File	Packages raw earthquake file into format usable to run on shake table using *q_data_q8*.
make_sine	Matlab M-File	Constructs compound sine waves that are usable in *q_data_xy_q8*.
power_spectrum	Matlab M-File	Computes the power spectrum of a signal.
setpoint_limit	Matlab M-File	Calculates the maximum amplitude and frequency position command that can be handled by the Shake

File Name	Type	Description
		Table II.
setup	Matlab M-File	Calculates various parameters used in all supplied Simulink models.
setup_stii_configuration	Matlab M-File	Sets various ST II model parameters, amplifier limitations, maximum stroke, and sensor calibration gains.
stii_limits	Matlab M-File	Computes maximum velocity, force, and acceleration of load given its mass and the ST II actuator specifications.
dderiv	Matlab P-File	Parsed code version of the dderiv.m Matlab M-File above.
q_scale	Matlab P-File	Produces a scaled position trajectory given desired acceleration data. When ran on the ST II, the yielded acceleration match the given desired acceleration.
shake_table_II_lib.mdl	Simulink Library	Contains various Simulink subsystems used to build the Shake Table II Simulink models.
debounce_lib.mdl	Simulink library	Contains the *Debounce Switch* S-Function Simulink block.
debounce.dll	Dynamic Link List	These various files implement the S-Function *Debounce Switch*.
debounce.c	C Source	
debounce_wrapper.c	C Source	
HIK000.AT2	Text File	Sample Kobe earthquake acceleration data file.
SYL090.AT2	Text File	Sample Northridge earthquake acceleration data file.

Table 12: List of files provided in the "mdl" folder on the Shake Table II CD.

6.3. Shake Table II WinCon Library

Most of the subsystem blocks inside the Shake Table II Simulink Models supplied, such as the q_sine_q8.mdl shown in Figure 25, are linked to the Shake Table II WinCon Library pictured in Figure 26. The *shake_table_II_lib.mdl* is opened through Matlab as an ordinary Simulink diagram. If changes are made to any of the blocks inside the library, then they are propagated to the linked blocks of the Simulink diagrams.

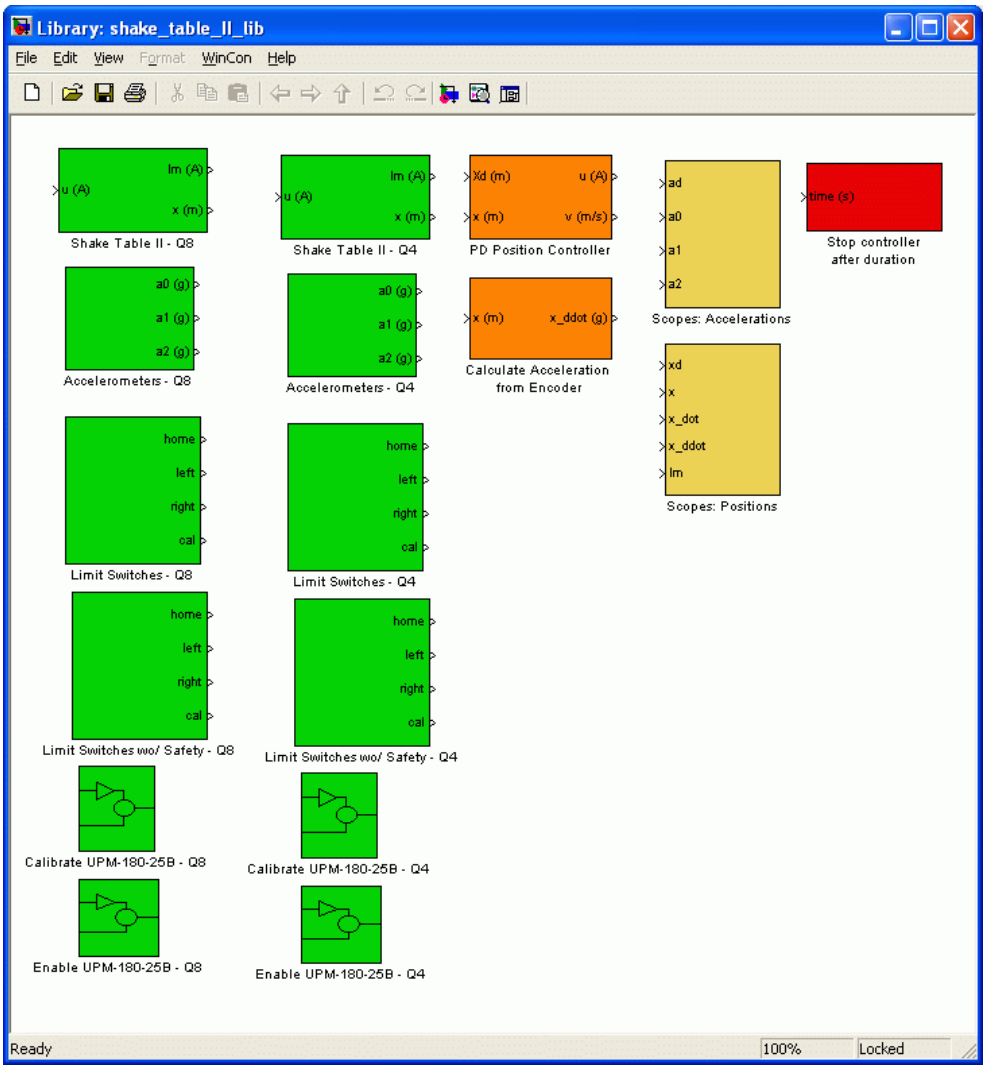


Figure 26: Shake Table II WinCon library.

The green subsystems on the left-hand side of the library interface with the actual shake table hardware using either the Quanser Q4 or Q8 hardware-in-the-loop board Simulink blocks. The middle orange blocks are calculation-based blocks, the yellow subsystems on the right contain scopes used to display data, and the red subsystem on the far right is used to stop a WinCon controller. This section describes each subsystem in the Shake Table II WinCon Library.

6.3.1. Shake Table II Subsystem

The *Shake Table II – Q8* and *Shake Table II – Q4* blocks interface with the ST II motor and encoder using the Q8 and Q4 board, respectively. The interior of the *Shake Table II – Q8* block is illustrated in Figure 27. Using the Quanser Q8 DAC Analog Output block, a current can be fed to the blue UPM-180-25B to drive the ST II motor. The current is saturated by the 25.0 A UPM current limit. The saturated current is then divided by the amplifier gain, 5 A/V, to convert the signal into a voltage (because the D/A channel of the DAC board outputs voltage) and so the commanded current, *u (A)*,

equals the current driving the motor. This voltage is passed through another saturation block with the Q4/Q8 board analog output limits, ±10V, and the resulting signal is passed to the Quanser Analog Output block. The input motor current is the saturated signal multiplied by the amplifier gain which is sent to the subsystem output variable *Im (A)*.

The Encoder Input block pictured in Figure 27 below outputs the number of counts measured by the shake table encoder, discussed in Section 3.2.5, which corresponds to the angular position of the ball-screw. The linear position of the stage is calculated by multiplying the number of counts times the encoder sensitivity gain, K_ENC.

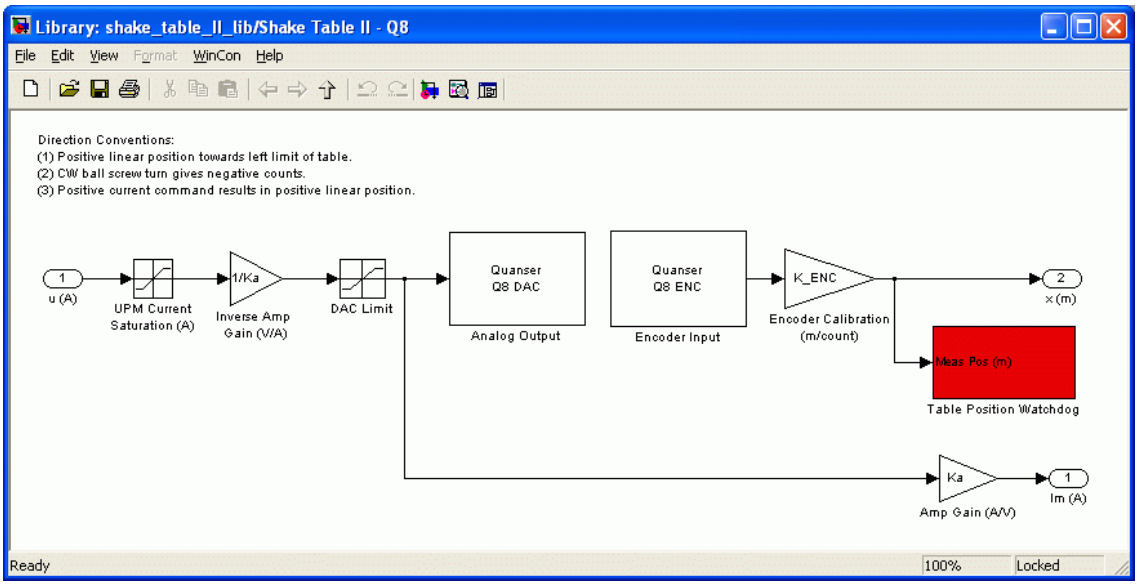


Figure 27: Shake Table II - Q8 subsystem.

The *Table Position Watchdog* block shown in Figure 28 stops the WinCon controller from running when the stage position measured equals or exceeds the P_MAX variable. This effectively stops the amplifier from driving the motor. By default, P_MAX is set to 7.62 cm, or 3.0 inches.

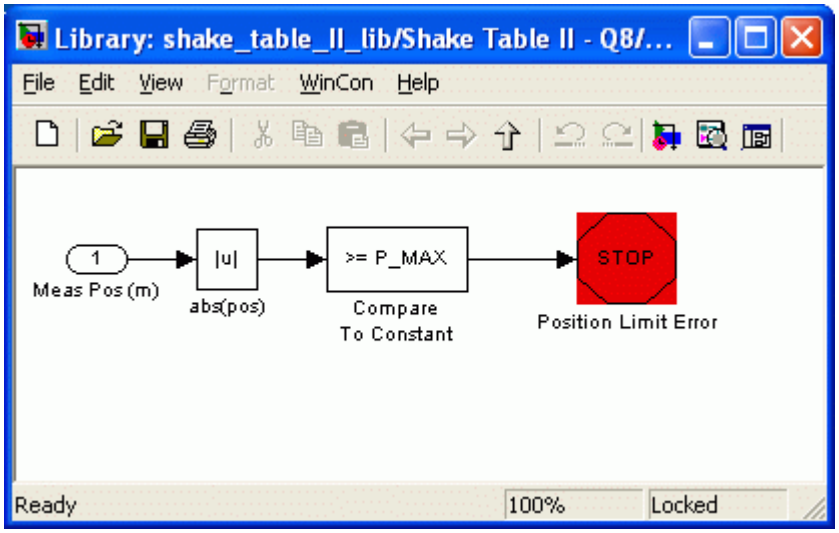


Figure 28: Software watchdog: stops WinCon when encoder reads over P_MAX variable.

6.3.2. Accelerometers Subsystem

As explained in Section 3.2.8, an accelerometer is mounted underneath the top stage of the Shake Table II and it is capable of measuring the acceleration in both the *x* and *y* directions. The *Accelerometers* block contained in the WinCon ST II Library and used in the *q_sine_zz*, *q_sweep_zz*, and *q_data_zz* Simulink diagrams is shown in Figure 29. The Analog Input block outputs three signals: the acceleration from Accelerometer #0 (already mounted on table), the acceleration Accelerometer #1 (may or may not be mounted), and the acceleration Accelerometer #2 (may or may not be mounted). As illustrated in Figure 29, the accelerometer signals are fed to *ACC 0: Bias and Filter, ACC 1: Bias and Filter,* and *ACC 2: Bias and Filter* blocks.

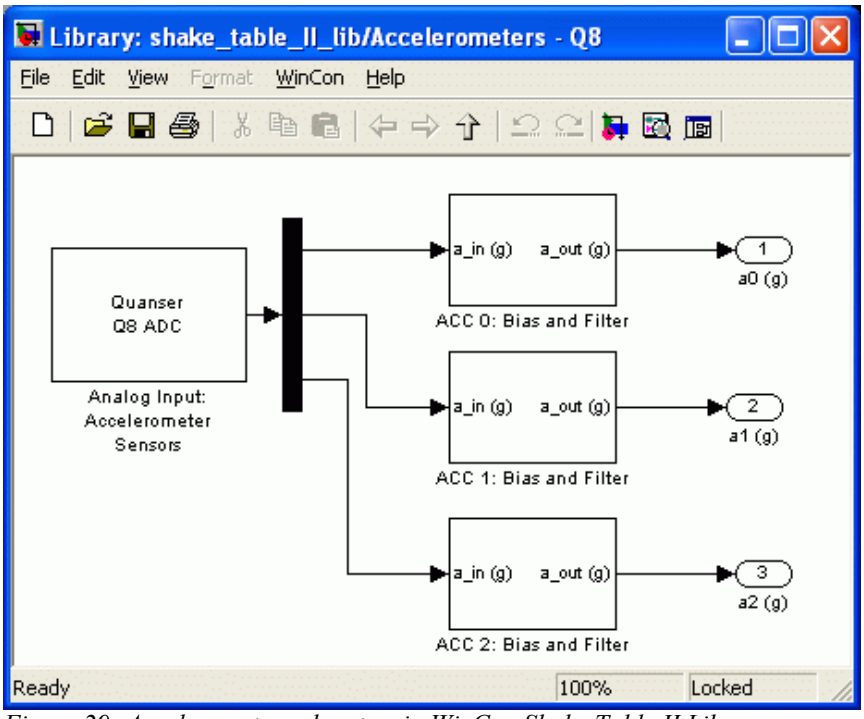


Figure 29: Accelerometer subsystem in WinCon Shake Table II Library.

The inside of the *ACC 0: Bias and Filter* block is shown in Figure 30. The *Bias Removal* block removes any initial non-zero measurement in the measured acceleration to ensure the readings are zero before beginning the experiment. In order to remove some noise in the analog acceleration signal, the resulting biased signal is passed through a second-order filter. The parameters for these filters are set in the *setup.m* script. Finally, in order for the accelerometer to be synchronized with the encoders, i.e. give positive measurements on positive encoder counts, the *Acceleration Calibration (g/V)* gain is set to the K_ACC parameter. The accelerometer calibration gain, K_ACC, is set in the *setup.m* file.

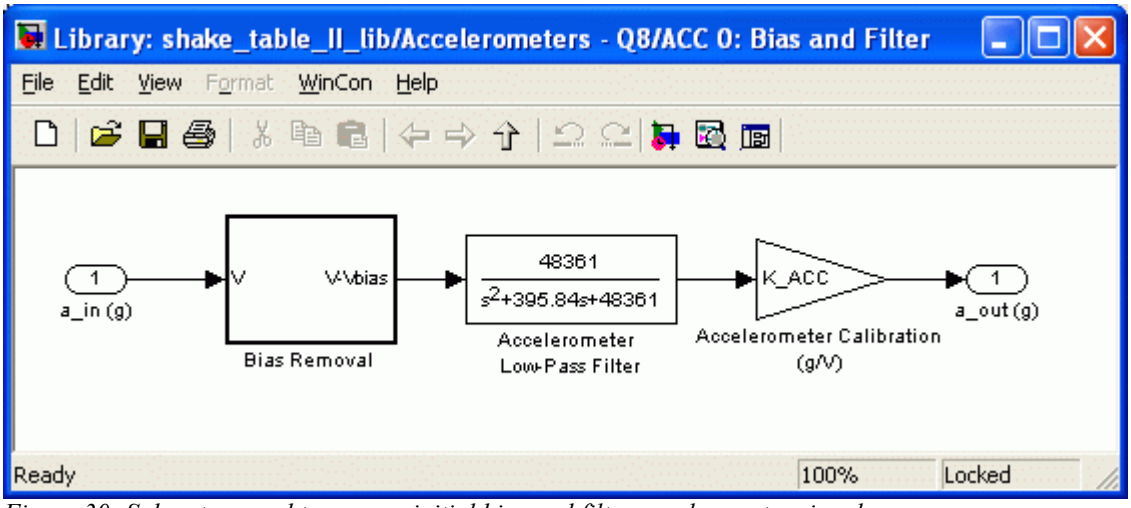


Figure 30: Subsystem used to remove initial bias and filter accelerometer signal.

6.3.3. Limit Switches Subsystem

The interior of the *Limit Switches – Q8* block is given in Figure 31. Using the *Stop with Error* block from the Quanser Toolbox, WinCon stops running the controller when either the *Left* or *Right* limit switch on the shake table is triggered. This is in addition to the PIC-based safety features in the UPM that also stop the amplifier when the *Left* or *Right* signals are activated. See Reference [2] for more information on the safety features of the UPM-180-25B device.

The digital readouts of the limit sensors are read using the *Quanser Digital Input* block shown in Figure 31. The micro-vibrations from the ball-screw and the high-frequency switching of the amplifiers motor leads can cause the limits sensors to be triggered unexpectedly. That is, they can output a high signal even though the table has not reached its maximum travel distance. As a result, the limit sensor signals are passed through a digital debounce system.

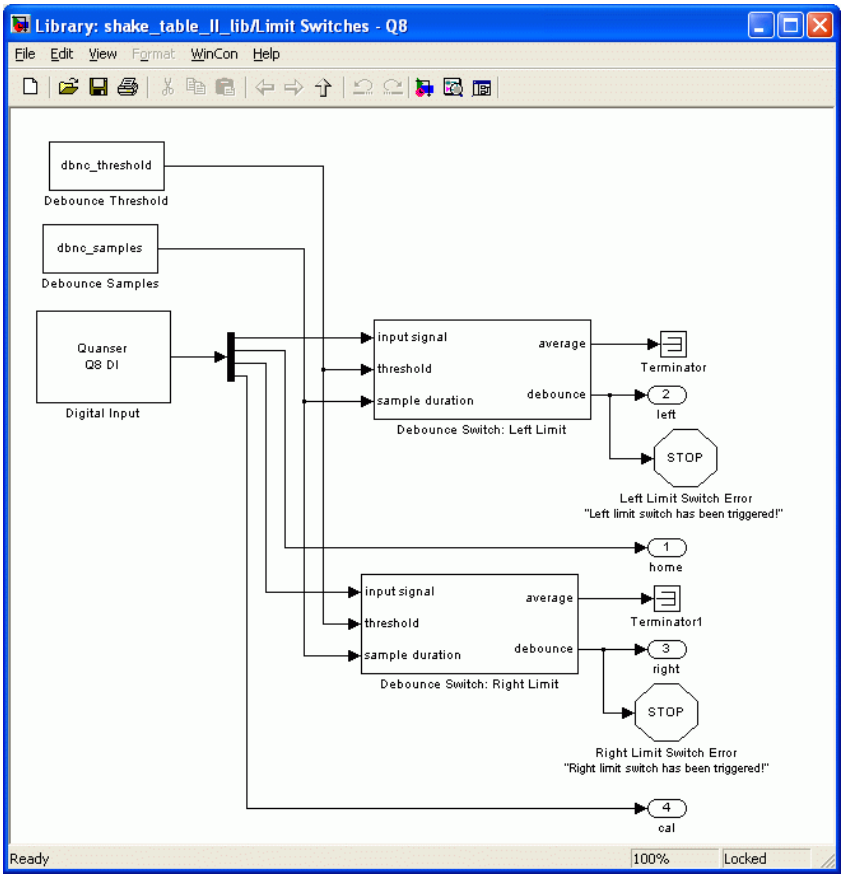


Figure 31: Limit switches subsystem in WinCon ST II Library.

The *Debounce Switch* block keeps track of the last *sample duration* samples of the *input signal*. For example, if the *sample duration* is 100 then the debounce system stores the last 100 samples of the *input signal*. The average of these samples is compared with the *threshold* input. If the average of the samples is larger than the threshold the *debounce* output is set to 1, otherwise it is set to 0. Figure 32 illustrates the debounce operation and how it ignores some of the noise. In this illustration, the *sample duration* is 100 samples and the sampling rate of the controller is 1 kHz. Thus the debounce averages

the last 1 millisecond of input signal data.

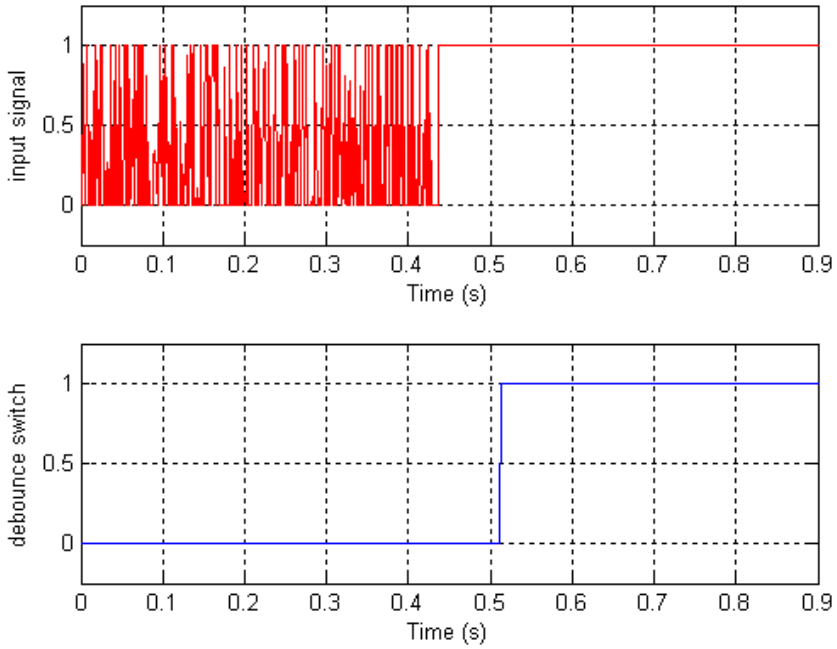


Figure 32: Demonstrating the digital debounce switch.

Also included in the *WinCon ST II Library* is the *Limit Switches wo/ Safety* block, shown in Figure 33, that is used when the calibrating the stage of the shake table. When calibrating, the *Left* and *Right* sensors are used to position to the stage to the mid-stroke position and, as a result, it is not desired to deactivate the amplifier when these signals are triggered. The *Limit Switches wo/ Safety* subsystem is identical to the *Limit Switches* block except there are no *Stop with Error* blocks that stop running WinCon when the *Left* or *Right* signals get activated.

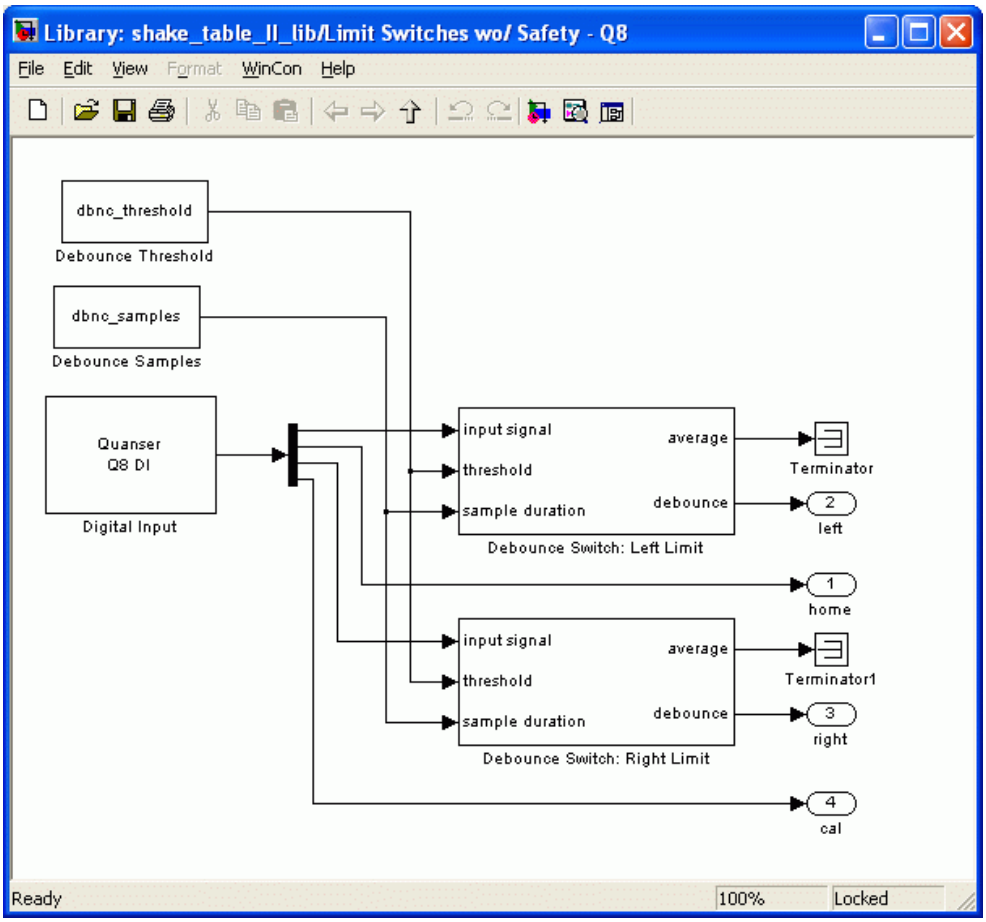


Figure 33: Limit Switches wo/ Safety subsystem in WinCon ST II library.

6.3.4. Enable Mode Subsystem

When using the Q8 board, the *Enable UPM-180-25B-Q8* block is required to enable the power amplifier in order to drive the ST II motor and therefore perform experiments (use the *Enable UPM-180-25B-Q8* for the Q4 card). The amplifier is therefore not enabled unless WinCon is running a controller that was built from a Simulink model that included this block. See Reference [2] for details on enabling the UPM-180-25B device.

6.3.5. Calibration Mode Subsystem

When using the Q8 board, the *Calibrate UPM-180-25B-Q8* block is required to place the UPM-180-25B device in *calibration* mode in order to center the stage of the shake table. When in the *calibration* mode, the amplifier is enabled and remains enabled even if the *Left* or *Right* limit switch is triggered (unlike in the *Enable* mode). It becomes deactivated, however, when the *Home* proximity sensor goes on. See Reference [2] for details on the UPM-180-25B calibration mode.

6.3.6. PD Position Control Subsystem

The *PD Position Control* subsystem shown in Figure 34 implements the proportional-derivative feedback loop depicted in Figure 35. It calculates the motor input current needed to move the stage to the desired position.

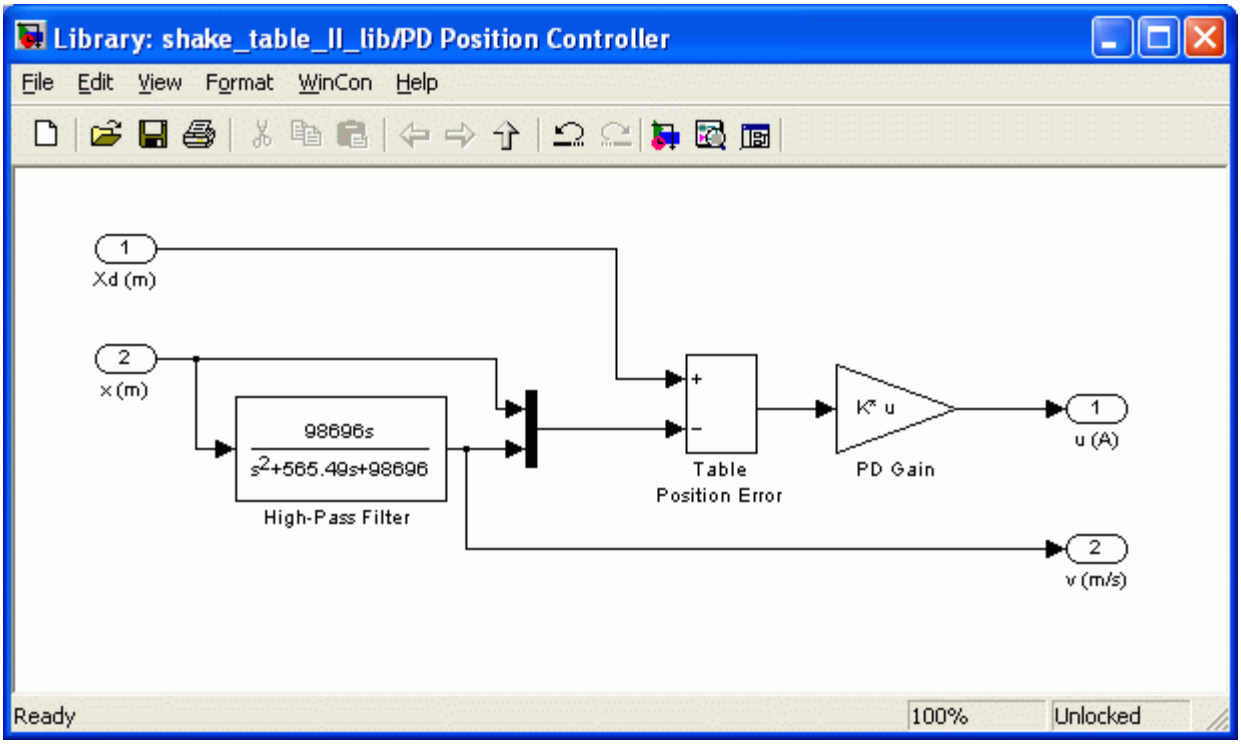


Figure 34: Proportional-derivative controller subsystem in ST II WinCon Library.

As illustrated under the *Shake Table II Plant* heading in Figure 35, the transfer function that describes the transition between the current applied to the motor, I_m, and the position, x, is

$$X(s) = \frac{I_m(s)}{K_f s^2}, \qquad [1]$$

where s is the Laplace operator, $I_m(s)$ is the Laplace transform of the motor current, X(s) is the Laplace representation of the stage position, and K_f is the open-loop gain. The open-loop model parameter is described

$$K_f = \frac{M_t P_b}{K_t} \qquad [2]$$

where, as specified in Table 4, M_t is the total mass being moved by the motor, P_b is the ball-screw pitch, and K_t is the motor current-torque specifications. To control the position of the stage a proportional-derivative, or PD, control scheme is used. The PD controller is illustrated in Figure 35 and can be described by the transfer function equation

$$I_m(s) = -k_p\,(X(s) - X_d(s)) - k_d\,(s\,X(s) - s\,b_{sd}\,X_d(s)) \qquad [3]$$

where $X_d(s)$ is the Laplace of the desired motor position (i.e. the setpoint), k_p is the proportional gain, k_d is the derivative gain, and b_{sd} is the velocity set-point weight. Substituting the PD controller given in Equation [3] into the open-loop model, Equation [1], and solving for $X(s)/X_d(s)$ results in the closed-loop transfer function of the system

$$\frac{X(s)}{X_d(s)} = \frac{k_p + k_d s\, b_{sd}}{K_f s^2 + k_p + k_d s} \quad [4]$$

The closed-loop transfer function describes how the stage position responds to a given position command.

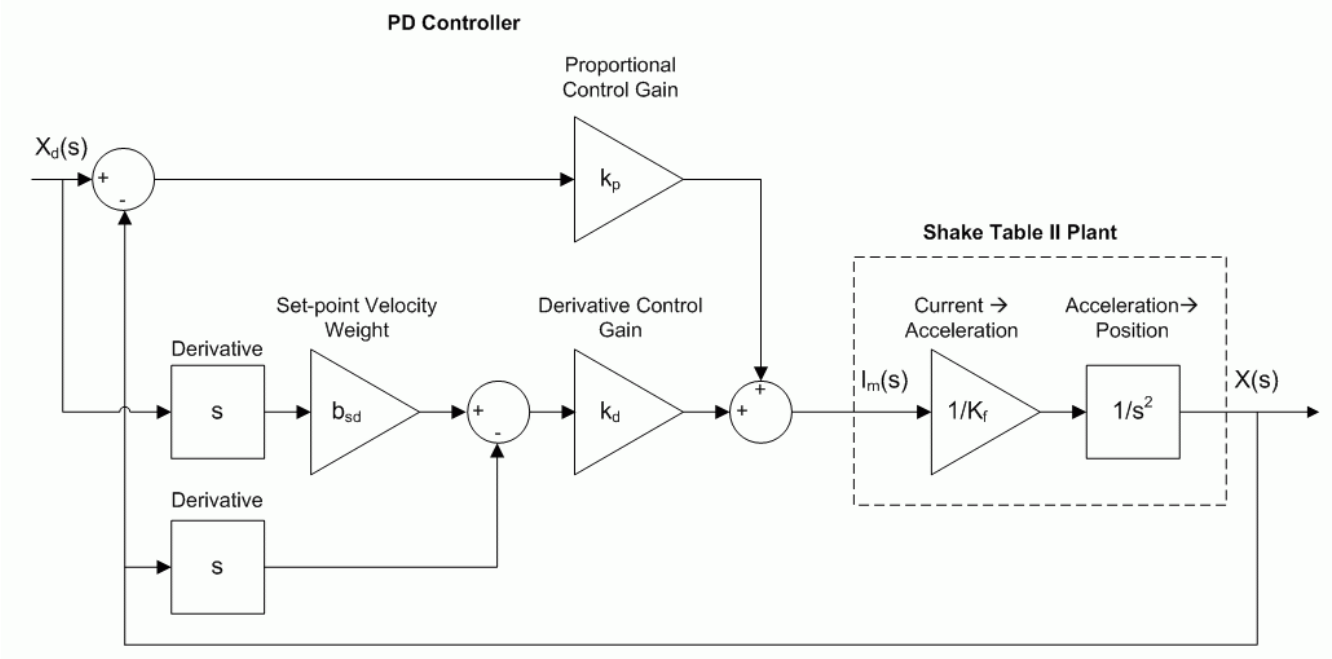


Figure 35: Block diagram of the control system used to position of ST II stage.

The PD controller implemented in the *PD Position Controller* block, shown in Figure 34, is structured as follows

$$u = K(X_d - X) \quad [5]$$

where K is the control gain, X_d is the setpoint state, and X is the measured state. The control gain vector is defined

$$K = [k_p, k_d]^T, \quad [6]$$

and the setpoint state

$$X_d = [x_d, v_d]^T \quad [7]$$

includes the desired stage position x_d, along with the desired stage velocity v_d. The desired position and velocity are generated via Simulink blocks as explained later in sections 6.5, 6.6, and 6.7. The state of

the system is defined

$$X = [x, v_x]^T \quad [8]$$

where x is the measured stage positions and v_x is the velocity of the stage. There is no sensor measuring the velocity of the Shake Table II stage directly, e.g. such as with tachometer. It is therefore computed by taking the derivative of the measured position and then filtering the result to eliminate noise. Effectively, the velocity is calculated using a second-order high-pass filter of the form

$$\frac{V_x(s)}{X(s)} = \frac{\omega_d{}^2}{s^2 + 2\,\zeta_d\,\omega_d\,s + \omega_d{}^2} \quad [9]$$

where ζ_d is the damping ratio of the filter and ω_d is the cutoff frequency of the filter (in rad/s). These filter parameters are set in the *lpf_cutoff_freq* Matlab M-File and in effect change the shape and bandwidth of the velocity response.

6.3.7. Acceleration from Encoder Subsystem

The interior of the *Calculate Acceleration from Encoder* ST II library block is shown in Figure 36. This system computes the acceleration of the stage as opposed to measuring its acceleration directly using the accelerometer sensor. This method may be used if the table is not equipped with an accelerometer or if under some circumstances this acceleration seems to yield better results (perhaps it has less noise when the stage is tracking a certain signal).

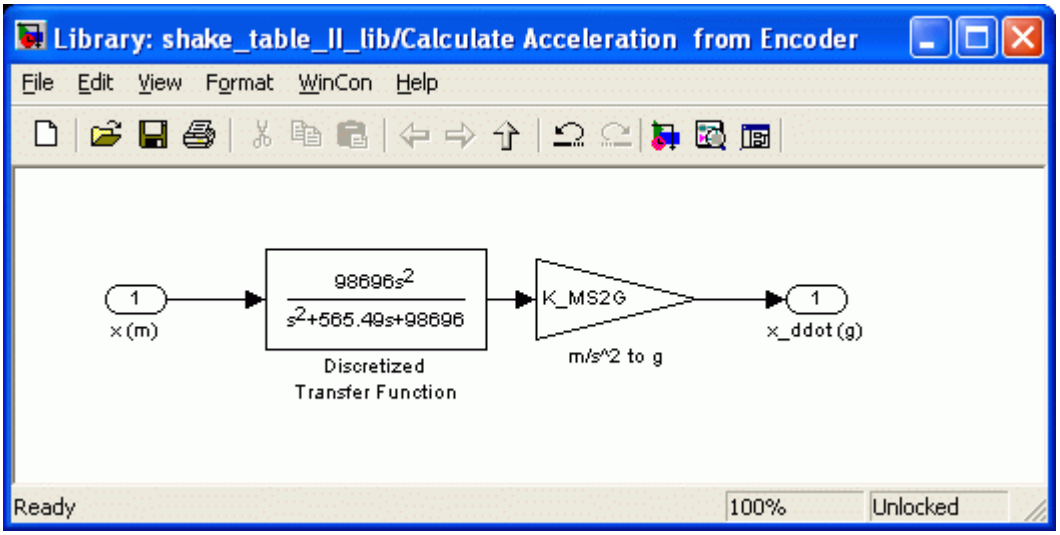


Figure 36: Calculate acceleration from position measurement subsystem in ST II WinCon Library.

The acceleration is calculated by taking the double-derivative of the position measurement and then passing that result though a low-pass filter. This is effectively done by taking the following transfer function of the stage position

$$\frac{A_{x,\,enc}(s)}{X(s)} = \frac{\omega_f{}^2 s^2}{s^2 + 2\,\zeta_f \omega_f s + \omega_f{}^2}, \qquad [10]$$

where ζ_f is the damping ratio of the filter and ω_f is the cutoff frequency of the filter. These filter parameters are set in the *lpf_cutoff_freq* Matlab M-File. The $A_{x,enc}(s)$ variable is the Laplace transform of the resulting acceleration. The acceleration calculation is initially in m/s^2 but, as shown in Figure 36, is converted to gravitational units, *g*, using the K_MS2G parameter.

6.3.8. Scope Subsystems

The *Scopes: Positions* subsystem contains scopes based on the stage position in metric and imperial units. For example, the *x (in)* plot displays the desired and measurement position of the Shake Table II stage in inches. The *Scopes: Accelerations* blocks has scopes that displays the acceleration data from up to three accelerometers. See Table 10 for a description of all these scopes.

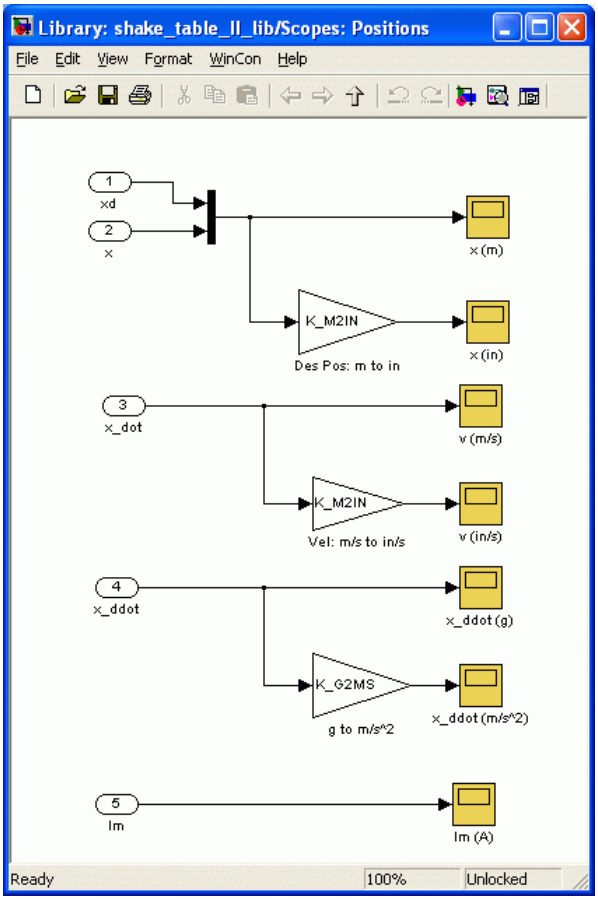


Figure 37: Positions scopes subsystem in WinCon ST II Library.

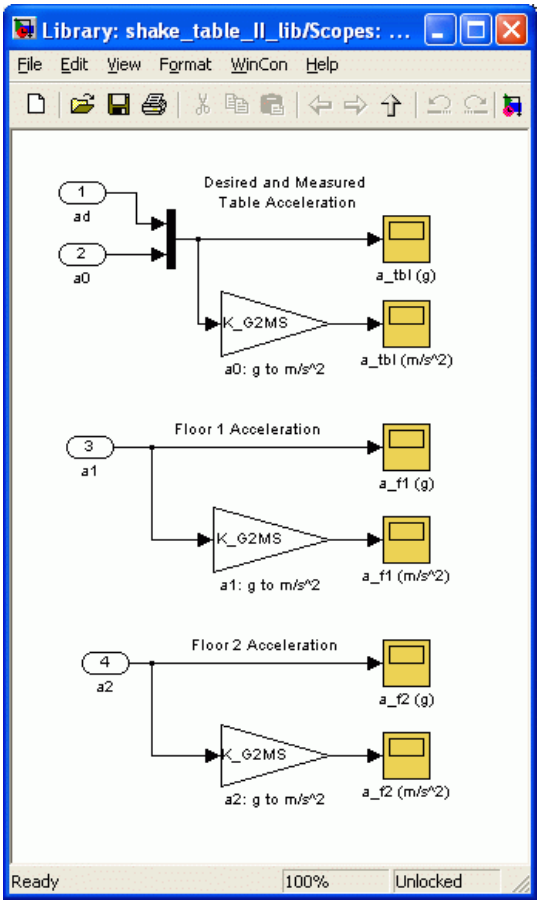


Figure 38: Acceleration scopes in WinCon ST II Library.

6.3.9. Stop after Duration Subsystem

The interior of the *Stop controller after duration* subsystem is depicted in Figure 39. This block is used in the *q_data_zz* Simulink diagram when using a predefined trajectory. The duration or length of the trajectory is stored in the *tf* parameter. When the running time of the associated WinCon Controller exceeds the *tf* variable, the *Stop With Error* block stops the controller from running and prompts the message "Controller duration reached." message to the user. The *Stop With Error* Simulink block is located in the Quanser Toolbox.

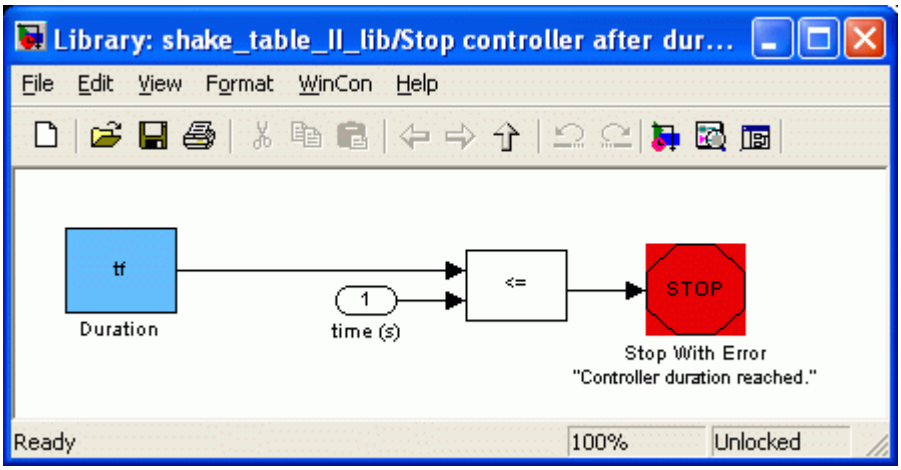


Figure 39: Stop controller after duration subsystem in WinCon ST II Library.

6.4. Matlab Scripts

There are five major scripts that are depicted in Figure 40 by the shaded square boxes: *setup.m*, *make_quake.m*, *make_sine.m*, *fft_eval_pos.m*,and *fft_eval_acc.m*. Figure 40 depicts the various file dependencies and calls made between many of the Matlab script files supplied. The *setup.m* is the main Matlab script that needs to be ran before building a WinCon Controller from any of the Shake Table II Simulink diagrams and before running other scripts such as *make_quake.m*. As described in Section 6.4.1, *setup.m* calls various Matlab script functions to load the ST II system parameters, calculate the position control gains, compute velocity and acceleration limits, set filter parameters, and so on.

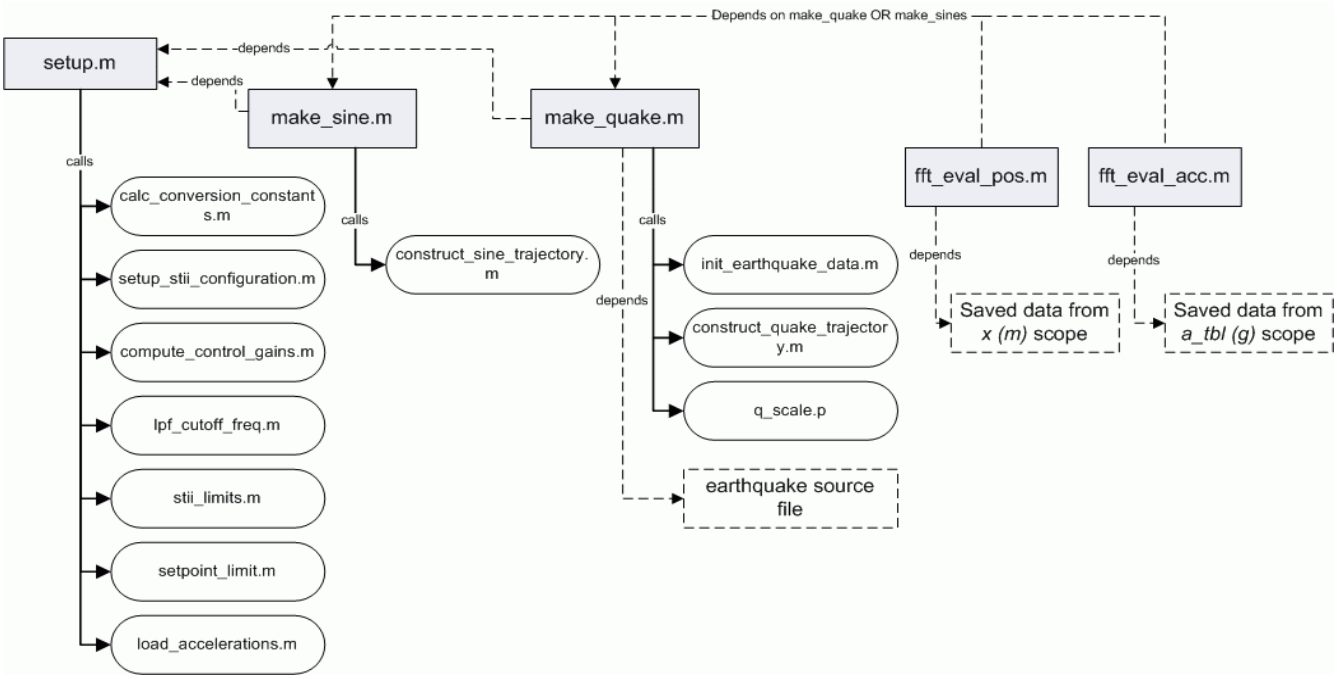


Figure 40: Dependencies and various M-File function calls.

The *make_sine.m* and *make_quake.m* scripts generate trajectories that can are used with the *q_data_zz* Simulink model. As described in Section 6.4.2, the *make_sines.m* script is used to generate a compound sine wave. Section 6.4.3 describes how to use the *make_quake.m* M-File to replay an earthquake on the shake table. This script needs a raw earthquake data file and calls several Matlab scripts, listed in Figure 40, to construct a trajectory. The *fft_eval_pos.m* script generates a Bode plot that compares the desired position of the shake table and the resulting measured position of the shake table after running an experiment. Similarly, the *fft_eval_acc.m* file plots the Bode of the desired and measured acceleration data. Both of these files need actual measured data stored in an MAT file. See Section 6.4.4 for details on how to use these scripts.

6.4.1. Setup Script: setup.m

The Matlab script *setup.m* has to be run before building any of the supplied Simulink models. It sets the parameters for the sensors, amplifiers, and controllers.

Follow these steps for instructions on how to run and configure the *setup.m* script:

1. Load Matlab.
2. Through the *Current Directory* window, go to the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).
3. Double-click on *setup.m* file to open it in the Matlab Editor window.
4. In the USER INPUT section shown in Text 1 below, the user can set the revision of the Shake Table II system, whether to use imperial or metric units, change the Universal Power Module type, and vary the maximum D/A output voltage of the data-acquisition board. By default and as depicted in Text 1, the script is configured for a Revision 4 table, the results are displayed in

metric units, the blue UPM-180-25B is used, and the maximum voltage of the Q4/Q8 is used. Users may wish to change *conversion* variable to display the results in either imperial or metric units.

```
% ************************************************************************
% USER INPUT
% ************************************************************************
% Enter revision of Shake Table II: REV1, REV2, REV3, or REV4.
STII_REV = 'REV4'; % 'REV1', 'REV2', 'REV3', or 'REV4'
% See results in IMPERIAL or METRIC.
conversion = 'METRIC'; % 'IMPERIAL' or 'METRIC';
% Enter type of UPM being used.
UPM_TYPE = 'UPM_180-25B';
% Maximum voltage deliverable by Q4/Q8 HIL (V)
VMAX_DAC = 10;
```

Text 1: The "USER INPUT" section in the setup.m script.

CAUTION: Most of these settings should not be changed. This script has been configured for the ST II system that was shipped and therefore most the parameters, with the exception of the conversion variable, should not be varied. In particular, do not change the STII_REV parameter unless you know for sure which revision your Shake Table II system is.

5. The CONTROL PARAMETERS section lets users specify the natural frequency and damping ratio of the position controller. See Section 6.4.1.1 for more information on the control gain design.

```
% ************************************************************************
% CONTROL PARAMETERS
% ************************************************************************
% Desired natural frequency of closed-loop system (Hz)
f0 = 15;
% Desired damping ratio of closed-loop system
zeta = 0.75;
```

Text 2: The "CONTROL PARAMETERS" section in the setup.m script.

CAUTION: Do not change these values unless you know what you are doing! For instance, increasing the natural frequency too much can result in vibrations that may harm the shake table device.

6. In the DEBOUNCE THRESHOLD section, users can change the threshold of the digital debounce system and number of samples used to take the mean of the input signal. See Section 6.3.3 for more information on the debounce S-Function used in *Limit Switches* subsystem.

```
% ************************************************************************
% DEBOUNCE THRESHOLD
% ************************************************************************
% Debounce is triggered when the average of the last 'dbnc_samples' is
% greater than the threshold.
dbnc_threshold = 0.8;
% Number of samples in input signal used in average calculation.
dbnc_samples = 250;
```

Text 3: The "DEBOUNCE THRESHOLD" section in the setup.m script.

7. Users can change the maximum amplitude of the sine sweep signal used in the *q_sweep_zz* Simulink diagram by varying the SWEEP_MAX parameters shown in Text 4. By default the amplitude of the sweep is limited to 4 mm.

```
% ************************************************************************
% MAX SWEEP AMPLITUDE
% ************************************************************************
% Maximum amplitude of sine sweep (m)
SWEEP_MAX = 4e-3;
```

Text 4: The "MAX SWEEP AMPLITUDE" section in the setup.m script.

CAUTION: Unless the end frequency of the sine sweep is lowered from the 15 Hz default, do not increase the SWEEP_MAX variable.

8. Run the Matlab M-File by clicking on Debug | Run in the Editor menu bar or clicking on the *Run* icon in the Editor tool bar.
9. As shown in Text 5, the script prompts the user for the mass of the load that is added to the top stage. Enter the mass of the payload in kilograms or pounds (depending on how the script is configured). If nothing is added, type 0 or simply press the ENTER key. Text 5 shows the typical output of the script after it is ran.

```
Enter any additional load on the top stage of the table (kg): 0

ENCODER CALIBRATION
   K_ENC = 1.55e-006 m/counts

LOAD
   Mass of top stage and bearing parts = 7.74 kg
   Load added = 0 kg
   Total load = 7.74 kg
   NOTE: Shake Table II specified for moving 15 kg at 2.5 g

LIMITS
   Position limit of table = +/- 76.2 mm
   Max velocity deliverable by motor = 664.9 mm/s
   Max force deliverable by motor = 708.661 N
   Max load acceleration = 2.5 g

Do you want to view the setpoint limitations plot? (y/[n]) y
```

Text 5: Output in Matlab Command Window after running setup.m script.

The encoder calibration gain, K_ENC, which is used to calculate the linear position of the stage from the measured encoder counts is displayed. Next the preload, added load, and total amount of mass that is being moved by the motor is summarized. Finally, the maximum stroke, velocity, force, and acceleration is displayed.

10. The script also gives the user the option to plot the setpoint limitations plot shown in Figure 41. Enter 'y' to view the plot. To skip the plot, either press the ENTER key or enter 'n'.

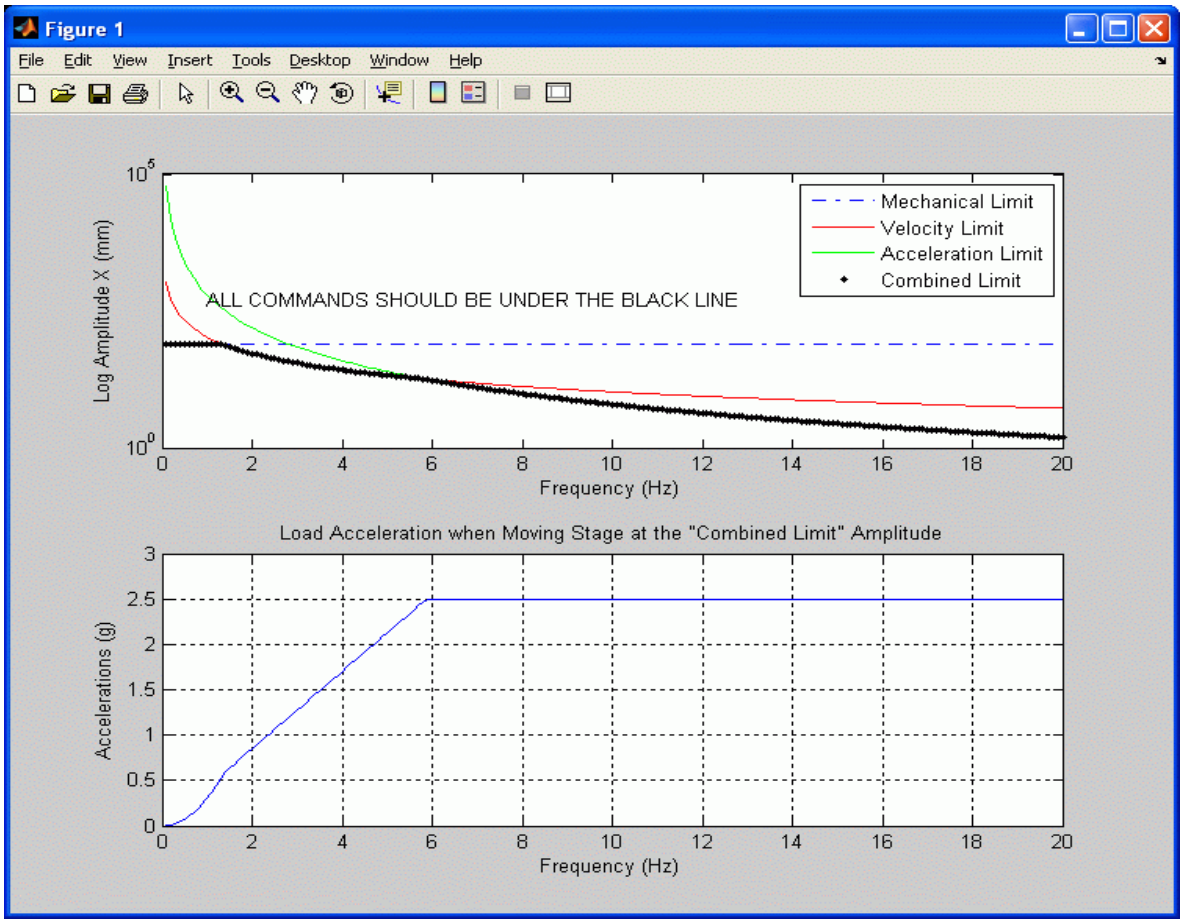


Figure 41: For a 0 kg load, this plot illustrates the maximum setpoint amplitude over a range of frequencies.

For a range of frequencies, the top plot in Figure 41 shows the maximum sine wave amplitude that can be commanded to the table for the stage to track. This plot takes the position, velocity, and acceleration limits of the Shake Table II system into account. The dash-dot blue line is the mechanical position limit of stage, the red line is the limit due to velocity, the green line is the limit due to acceleration, and the black line is the combined limit. Low frequency commands in 1.0-1.25 Hz range are limited due to the table travel. When in the 1.25-6.0 Hz range, the amplitude is constrained by the velocity limitations of the table. For higher frequencies, the command is constrained by the imposed acceleration limitation of the Shake Table II. All sine commands to the table should fall under the black line. For instance, when tracking a sine wave with a frequency of 8 Hz the user should not command an amplitude that exceeds 8.4 mm. The bottom plot shows the acceleration of the load when the stage is tracking a sine wave at varying

frequencies with an amplitude specified by the combined limit. For example, when running a sine wave at 4 Hz with an amplitude 25.7 mm, the load would reach accelerations of 1.68 g. See Section 6.4.1.2 for details on generating this plot.

6.4.1.1. Control Gain Design: compute_control_gains.m

The control gains are calculated in the M-File called *compute_control_gains.m*. This function is called by *setup.m* to design the proportional control gain, k_p, and the derivative control gain, k_d, based on the ST II model parameters, the load mass, and the control specifications. Table 13 lists some sample control gains generated by the script for various added load mass, M_l. The maximum load that can be added to the stage to achieve the rated ST II acceleration of 2.5 g is 7.26 kg.

M_l (kg)	M_t (kg)	f_0 (Hz)	ζ	k_p (V/m)	k_d (V.s/m)
0.0	7.74	15.0	0.75	2425.4	38.6
2.5	10.24	15.0	0.75	3208.8	51.1
5.0	12.74	15.0	0.75	3992.2	63.5
7.26	15.0	15.0	0.75	4700.4	74.8

Table 13: Sample controls gains calculated for a varying load mass.

The total mass being moved by the motor is denoted by the M_t variable. There are two design control parameters that are used to generate the control gains: the natural frequency, f_0, and the damping ratio, ζ. Generally speaking, the natural frequency determines the speed of the response and the damping ratio determines the shape of the response (i.e. the overshoot). In order to satisfy these specifications, both controls gains increases as the load mass is augmented. The control design is explained next to give some background on how these gains are generated.

The closed-loop transfer function that describes the response of the stage given a desired position was developed earlier and given in Equation [4] in Section 6.3.6. It is a second-order system and when the set-point velocity parameter $b_{sd} = 0$, it can be mapped to the general second-order transfer function

$$H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \qquad [11]$$

where ω_0 is the natural frequency and ζ is the damping ratio. Note that ω_0 is the natural frequency in radians per second while f_0 is the natural frequency in Hertz. The relation between the two is

$$\omega_0 = 2\pi f_0 . \qquad [12]$$

The denominator of the Shake Table II closed-loop transfer function in Equation [4] can be mapped to the denominator of transfer function [11], which is known as the characteristic equation, by setting the controls gains to

$$k_p = \omega_0^2 K_f \quad [13]$$

and

$$k_d = 2\zeta\omega_0 K_f . \quad [14]$$

The gains listed in Table 13 are generated by *setup.m* using the proportional gain relationship [13] and the derivative gain formula defined in [14].

6.4.1.2. Computing the Maximum Setpoint: setpoint_limit.m

The *setup.m* script gives the option to generate the *maximum setpoint* plot. When the stage of the Shake Table II is tracking a sine wave, the maximum amplitude and frequency of the waveform must be known. By taking into account the Shake Table II limits, the *maximum setpoint* plot illustrates the maximum amplitude of the sine wave position command when it is being run at a certain frequency.

Consider the sine wave position setpoint

$$x_d = A_d \sin(2\pi f t) \quad [15]$$

where A_d is the desired amplitude, *f* is the frequency, and *t* is the continuous time. The velocity of the command position is

$$v_d = 2A_d \cos(2\pi f t)\pi f \quad [16]$$

and the corresponding acceleration is

$$a_d = -4A_d \sin(2\pi f t)\pi^2 f^2 . \quad [17]$$

The maximum sine wave amplitude that the ST II stage can track given a certain frequency depends the following constraints: the maximum stroke of the table, the maximum stage velocity, and the maximum acceleration. When starting at the center or *home* position, the stage is mechanically limited to moving ±3-inches. Therefore the maximum position of the table is

$$x_{max} = 0.07620\,[m], \quad [18]$$

as defined in Table 4.

The velocity and acceleration limits of the table are computed in the Matlab M-File called *stii_limits.m*. Given the back-emf parameter of the ST II motor, K_m, and the maximum output voltage of the UPM-180-25B, VMAX_UPM, the maximum angular rate of the Shake Table II motor equals

$$\Omega_{max} = \frac{VMAX\_UPM}{K_m} . \quad [19]$$

The linear velocity of the stage is therefore

$$v_{max} = \frac{1}{2}\frac{\Omega_{max} P_b}{\pi}, \quad [20]$$

where P_b is the ball-screw pitch. The maximum force that can be delivered by the actuator is

$$F_{max} = \frac{K_t \, IMAX\_UPM}{P_b}. \quad [21]$$

where IMAX_UPM is the maximum peak current of the power amplifier and K_t is the current-torque constant of the ST II motor. The maximum load acceleration depends on the load mass, M_t, and the rated acceleration of the table, A_LIM_RATED = 2.5 g. It is expressed

$$a_{max} = \min\left(\frac{F_{max}}{M_t}, A\_LIM\_RATED\right). \quad [22]$$

Evaluating equations [20] and [22] using the ST II parameters in Table 4 and the UPM-180-25B specifications given in Reference [2], the maximum velocity and acceleration are

$$v_{max} = 0.665\left[\frac{m}{s}\right]. \quad [23]$$

and

$$a_{max} = 24.5\left[\frac{m}{s^2}\right]. \quad [24]$$

Table 14 lists the equations to calculate the maximum setpoint amplitude due to the position limit $A_{max,p}$, due to the maximum velocity $A_{max,v}$, and due to the acceleration constraint $A_{max,a}$.

Limit	*Maximum Sine Wave Amplitude*
Position	$A_{max,p} = x_{max}$
Velocity	$A_{max,v} = \frac{1}{2}\frac{v_{max}}{\pi f}$
Acceleration	$A_{max,a} = \frac{1}{4}\frac{a_{max}}{\pi^2 f^2}$
Combined	$A_{max} = \min\left(x_{max}, \frac{1}{2}\frac{v_{max}}{\pi f}, \frac{1}{4}\frac{a_{max}}{\pi^2 f^2}\right)$

Table 14: Equations used to find the maximum setpoint amplitude.

With the equations given in Table 14, the top plot shown in Figure 41 can be generated. The bottom plot displays the acceleration of the load when the stage is tracking a sine wave at various frequencies at the amplitude specified by the combined limit. It is calculated in the Matlab script called *load_accelerations.m* with the equation

$$a_{max} = 4 A_{max,\,a} \pi^2 f^2 \quad [25]$$

6.4.2. Generating Composite Sine Wave: make_sine.m

The Matlab script file called *make_sine.m* generates a compound sine waveform that can be used with the *q_data_zz* Simulink model. Given a set of sine wave amplitudes, for example $A_d = [A_1, A_2, A_3]$ and a corresponding set of frequencies, $f_d = [f_1, f_2, f_3]$, the script generates a time-based array with the sine wave position

$$x_d = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t) + A_3 \sin(2\pi f_3 t) \quad [26]$$

the velocity

$$v_d = 2 A_1 \cos(2\pi f_1 t)\pi f_1 + 2 A_2 \cos(2\pi f_2 t)\pi f_2 + 2 A_3 \cos(2\pi f_3 t)\pi f_3 \quad [27]$$

and the acceleration

$$a_d = -4 A_1 \sin(2\pi f_1 t)\pi^2 f_1{}^2 - 4 A_2 \sin(2\pi f_2 t)\pi^2 f_2{}^2 - 4 A_3 \sin(2\pi f_3 t)\pi^2 f_3{}^2 \quad [28]$$

Follow these steps to use the script:

1. Load Matlab.
2. Through the *Current Directory* window, go to the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).
3. Open the *make_sine.m* file.
4. The INPUT section of the *make_sine* script is shown in Text 6 below. Set the amplitude vector, A_d, and frequency vector, f_d, to create a desired compound sine wave. Generally speaking an amplitude of $A_d = [A_1, A_2, ..., A_n]$ and $f_d = [f_1, f_2, ... f_n]$ can be defined.

```
% ************************************************************************
% INPUT
% ************************************************************************
% sine wave amplitude for each excitation (mm)
Ad  = [2.5, 2, 0.5];
% axes sine wave excitation frequencies (Hz)
fd = [1, 4, 10];
% time duration (s)
tf = 3;
% sampling period (s/sample)
dt = default_sample_time;
```

Text 6: "INPUT" section in the make_sine.m script.

5. The MAKE SINE WAVE section of the *make_sines.m* script is shown in Text 7. The *construct_sine_wave_trajectory* file creates the position, velocity, and acceleration time-based array. Because the UPM-180-25B amplifier is not enabled for the first 0.7 seconds, the first second of the compound sine wave is always padded with zeros. The total duration of the waveform is therefore t_f+1 seconds.

```
% ************************************************************************
% MAKE SINE WAVE
% ************************************************************************
% construct sine wave trajectory
[t, xd_mm, vd_mm, ad_mm] = construct_sine_trajectory(fd, Ad, tf, dt);
% Desired position (m)
xd = xd_mm / 1000;
% Desired velocity (m/s)
vd = vd_mm / 1000;
% Desired acceleration (g)
ad = ad_mm / 1000 / 9.81;
```

Text 7: "MAKE SINE WAVE" section in make_sine.m script.

6. Run the Matlab M-File by clicking on Debug | Run in the Editor menu bar or clicking on the *Run* icon in the Editor tool bar.

WARNING: **Ensure the q_data_zz.mdl Simulink Model is open before running the *make_sine.m* script.** The sampling time set in q_data_zz.mdl, *default_sample_time*, is loaded into the Matlab workspace when the Simulink diagram is open. Otherwise a missing variable error message will be prompted.

7. After generating the sine wave, the *make_sine.m* script generates a plot. The plot pictured in Figure 42 was generated when using the A_d and f_d vectors specified in Text 6.

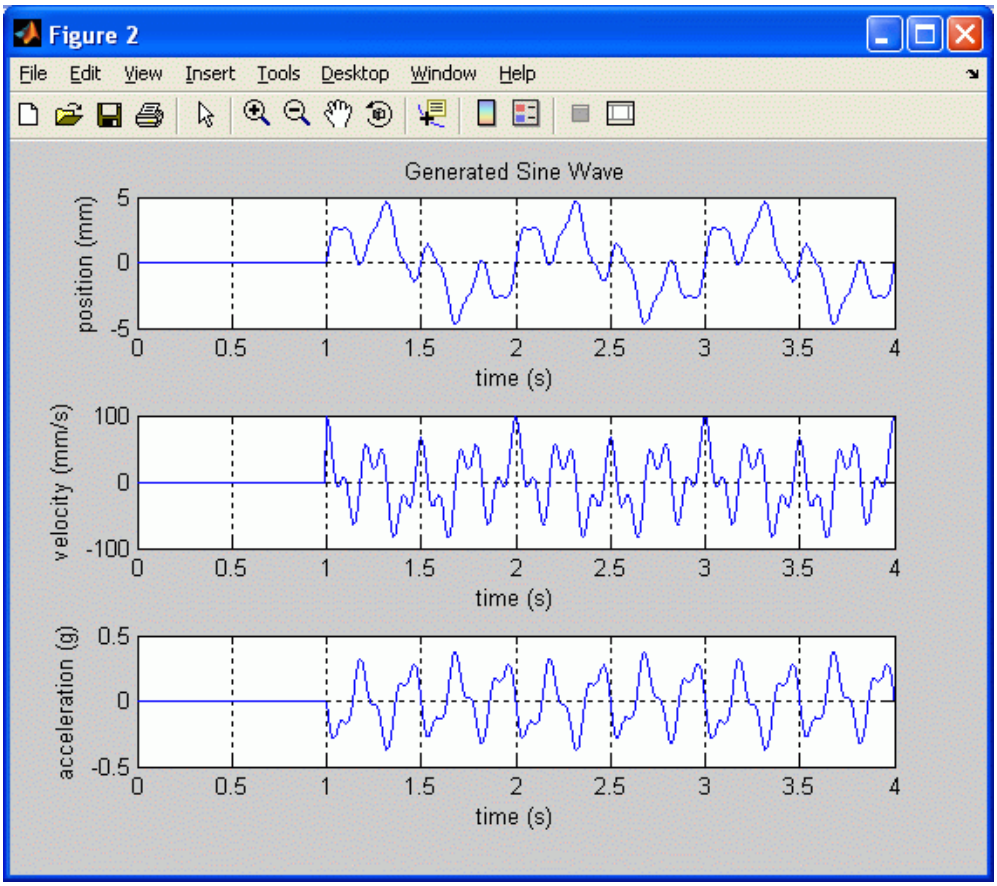


Figure 42: Composite sine wave plot generated by the make_sine.m script.

6.4.3. Constructing Scaled Earthquake: make_quake.m

The *make_quake.m* script builds a trajectory that can be used in the *q_data_xy_q8* Simulink diagram and ran on the shake table. The resulting trajectory created is the setpoint or command position that is to be tracked by the stage in order to achieve the same accelerations as the recorded earthquake. Section 6.4.3.1 describes how to run the make_quake.m script in order to do this.

As listed in Table 12, the Northridge, SYL090.AT2, and Kobe, HIK090.AT2, raw earthquake acceleration files are already supplied with the Shake Table II CD. Additional earthquake data files can be downloaded from the Internet from locations such as the Berkeley website, as explained in Section 6.4.3.2.

6.4.3.1. Running the Script

Follow these steps to run the make_quake.m file:

1. Load Matlab.
2. Through the *Current Directory* window, go to the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).

3. Open the *make_quake.m* file.
4. The INPUT section of the *make_quake* script is shown in Text 8 below. Enter the name of earthquake file that is to be replayed on the shake table. As shown in Text 8, the file is set to "HIK000.AT2" which is the Kobe earthquake. The *x_max* parameter determines the maximum position of the scaled setpoint trajectory.

```
% ************************************************************************
% INPUT
% ************************************************************************
% name of data source file, ex. 'HIK000.AT2' or 'SYL090.AT2'
input_filename = 'HIK000.AT2';
% Maximum scaled position (cm). NOTE: Set below 7.62 cm stroke limit.
x_max = 3.0;
```

Text 8: "INPUT" section in the make_sine.m script.

CAUTION: Do not set *x_max* greater to value greater then the maximum stroke of the table! Keep the variable under 7.0 cm.

5. The MAKE QUAKE section is shown in Text 9. As illustrated in Figure 43, there are three main steps to create the position setpoint, x_d, from recorded earthquake data.

```
% ************************************************************************
% MAKE QUAKE
% ************************************************************************
% load recorded acceleration earthquake data in workspace (g) and sample
% time (s)
[dt,acc_data] = init_earthquake_data(input_filename);
% construct trajectory
[t,a] = construct_quake_trajectory(acc_data,dt);
%
% Compute desired position of STII to achieve actual quake accelerations.
%   t       Setpoint time arrya (s).
%   xd_cm   Position setpoint array (cm)
%   ad      Desired acceleration array (g)
%   tf      Duration of the scaled earthquake trajectory (s)
[t, xd_cm, ad, tf] = q_scale(t, a, x_max);
% Convert desired scaled position (m)
xd = xd_cm / 100;
```

Text 9: "MAKE QUAKE" section in make_sine.m script.

The *init_earthquake_data.m* function extracts the sampling time information of the recorded earthquake, *dt*, and compiles the acceleration data from the AT2 file, which is the four column format, into an array called *acc_data*. The *construct_quake_trajectory.m* script creates a trajectory containing the recorded earthquake acceleration data, [*t*,*a*].

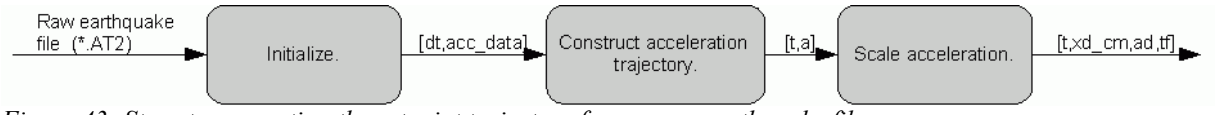


Figure 43: Steps to generating the setpoint trajectory from a raw earthquake file.

Then, *q_scale.p* goes through a scaling algorithm and outputs the time of the trajectory *t*, the

scaled position setpoint in centimeters *xd_cm*, the desired acceleration *ad*, and the duration of the tremor *tf*.

6. Run the Matlab M-File by clicking on Debug | Run in the Editor menu bar or clicking on the *Run* icon in the Editor tool bar.

   The output displayed in the Matlab Command Window should be similar to Text 10 below. The Kobe earthquake had a maximum displacement of 3.09 cm and this was scaled down to 3.0 cm (as set by *x_max*). In order to achieve the same acceleration as Kobe, the time of the generated trajectory is compressed from 77.98 to 76.82 seconds.

```
*** Usage : [Tc,Xc,Ac,Te]=q_scale(t,a,xmax)
t   : array of time at equal sampling intervals in seconds
a   : array of acceleration record in g to match t
xmax: maximum amplitude of motion that you want in cm
Note that this should not exceed the limits of the table !!
Tc  : Command time array
Xc  : Position command array that should be commanded to the table in cm
Ac  : Acceleration array in g that is the result of differentiating Xc
twice
Te  : Duration of the run

 Original time step:    0.02000
 Step 1 of 3: Get displacements
 Step 2 of 3: Scale records
 Ratio of table displacement to ground displacement: 0.970064

 Step 3 of 3: Scaling time
 Time step after scaling = 0.019698
*** Done ***
*** Displacement scaled from original movement of  3.09 cm to  3.00 cm
*** Time scaled from original duration of 77.98 seconds to 76.82 seconds
*** Record size is  3950 samples
```

Text 10: Output of make_quake.m script in Matlab Command Window.

7. The *q_scale.p* function script also generates a plot, which is pictured in Figure 44 for the Kobe earthquake, that displays the desired acceleration and recorded earthquake acceleration in the top plot (in gravitational units) and the scaled position setpoint in the bottom plot (in centimeters). Notice that the desired acceleration *ad*, which is computed from the scaled position setpoint *xd_cm,* is the same as the actual recorded acceleration of the earthquake, *a*. As a result, the two plots are layered on top of each other.

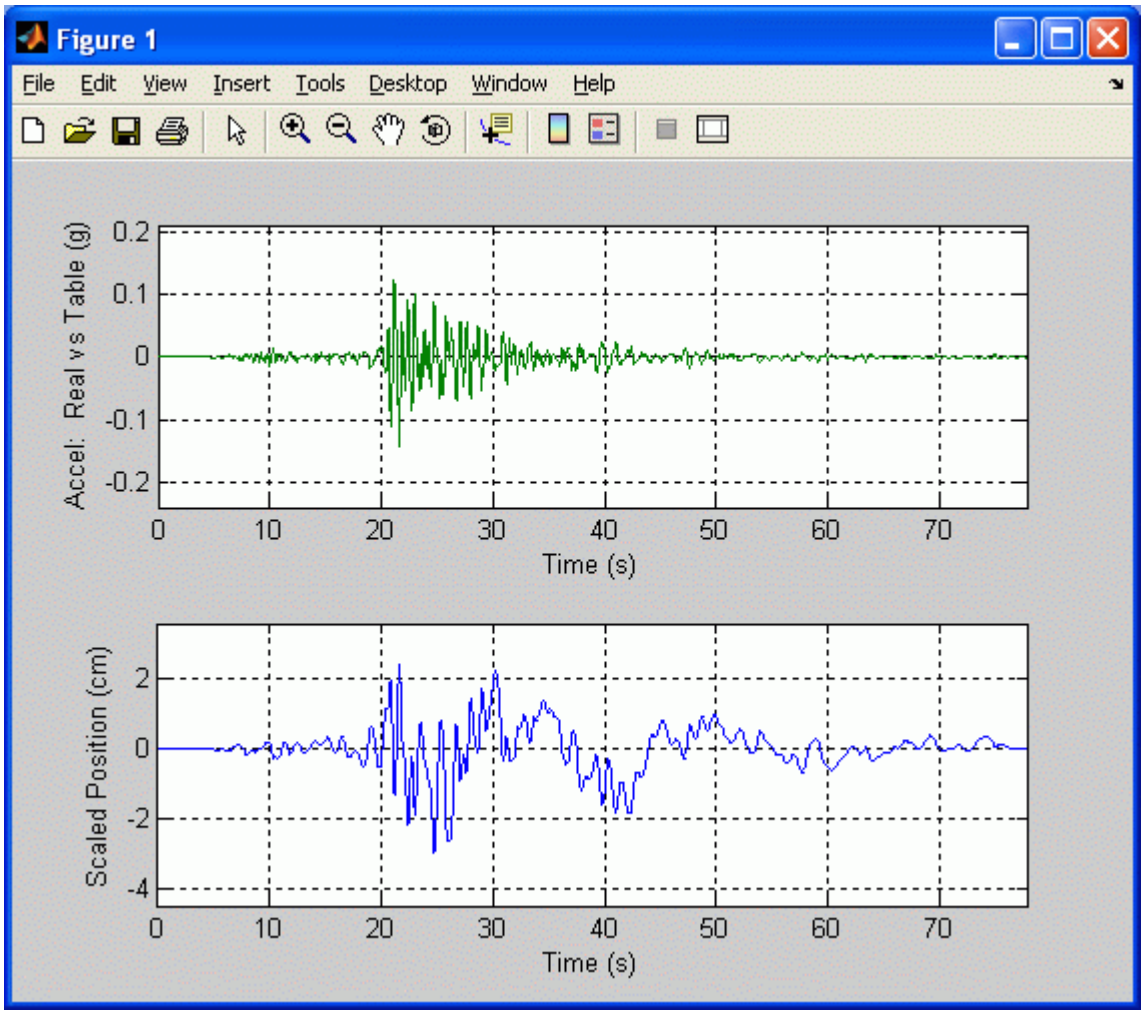


Figure 44: Plot generated by q_scale.p showing generated acceleration and scaled position.

Once the setpoint is generated, the q_data_zz Simulink Model can be used to replay the earthquake on the Shake Table II. See Section 6.7.2 for the procedure to run the tremor on the table. The next sections explain how to download additional earthquake data and generate Bode plots.

6.4.3.2. Downloading an Earthquake

There are a variety of resources on the Internet where real earthquake data can be downloaded. Two example sources are the Pacific Earthquake Engineering Research Center (PEER) Strong Motion Database website at http://peer.berkeley.edu/smcat/browse.html from the University of California and the Lamont-Doherty Earth Observatory of Columbia University at http://www.ldeo.columbia.edu/nceer/strongmo.html.

On the PEER website, each earthquake has various measurement stations and each station contains recorded displacement, velocity, and acceleration data of the tremor at different directions. Follow the procedure below to download a displacement record of the Kobe earthquake from the PEER database by *browsing*:

1. Open a browser and enter the address: http://peer.berkeley.edu/smcat/browse.html
2. As shown in Figure 45, look through the various earthquake records and find the Kobe earthquake data that was recorded at station HIK. Click on the HIK000 heading. Note that there are three directions recorded for the Kobe earthquake at this HIK station: HIK000, HIK090, and HIK-UP. For the Shake Table II, only the planar headings are of interest: HIK 000 and HIK 090, and not the up/down displacement of HIK-UP.

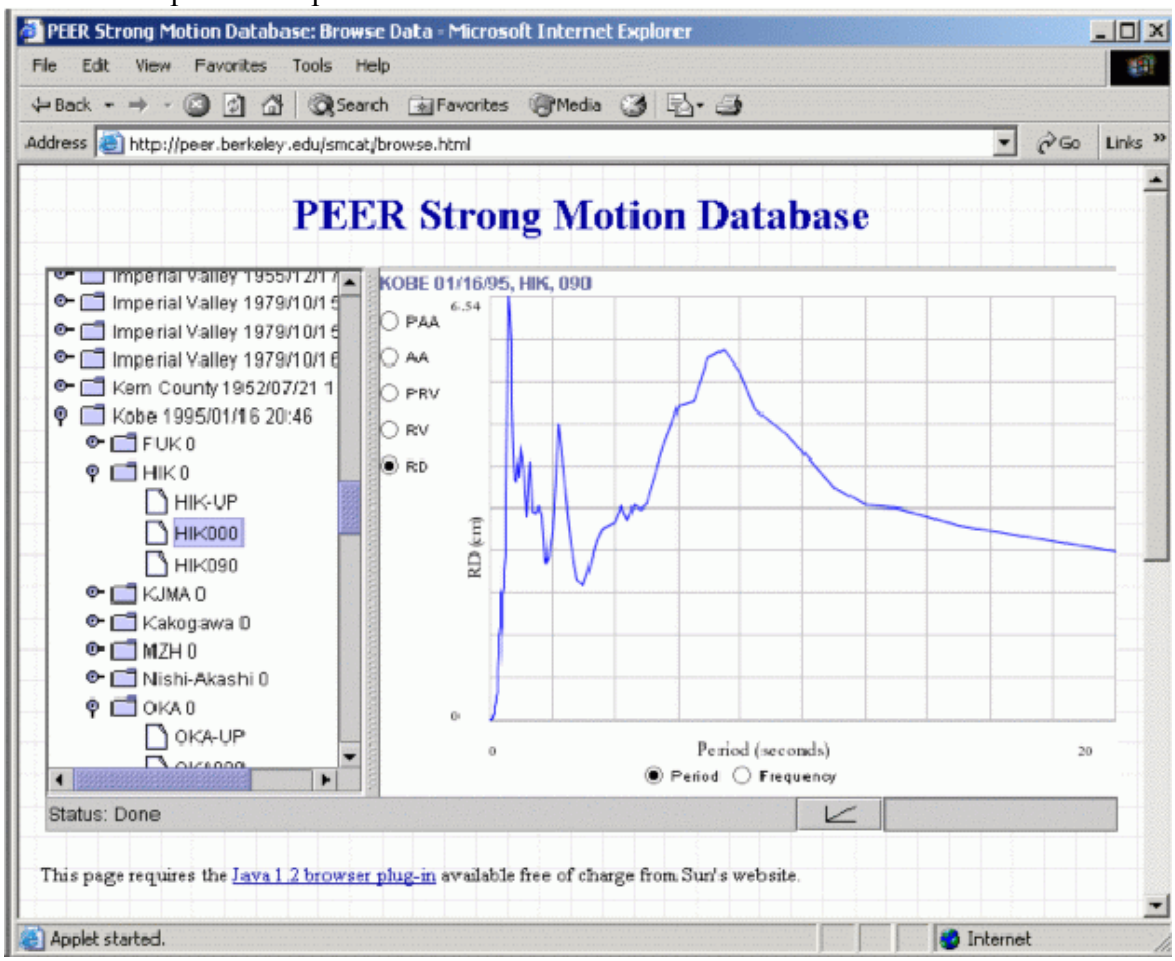


Figure 45: Browsing for an earthquake on the Berkeley website.

3. Select the *RD* option in the plot window, as shown in Figure 45, to display the relative displacement for the given heading.
4. Right-click on the earthquake record and select *Download...* from the drop-down menu.
5. The acceleration, velocity, and displacement recorded data is available to be downloaded. Choose the *Acceleration Time History* to be downloaded.
6. Once the earthquake record is downloaded it will be loaded into the browser. If using Internet Explorer it is loaded as an ASCII file. Save the file as a AT2 text file in the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD). The earthquake displacement record for one direction has been downloaded and can be used to command the shake table using *make_quake.m*.

Alternatively, an earthquake can be *searched* through the PEER website (as opposed to *browsing*). Here is the procedure to find the Kobe earthquake displacement data recorded at the HIK station:

1. Open an Internet browser and enter the following address to search for an earthquake: http://peer.berkeley.edu/smcat/search.html
2. As shown in Figure 46, select *Kobe 1995/01/16 20:46* under the *Earthquake* drop-down menu and click on the *Search* button.

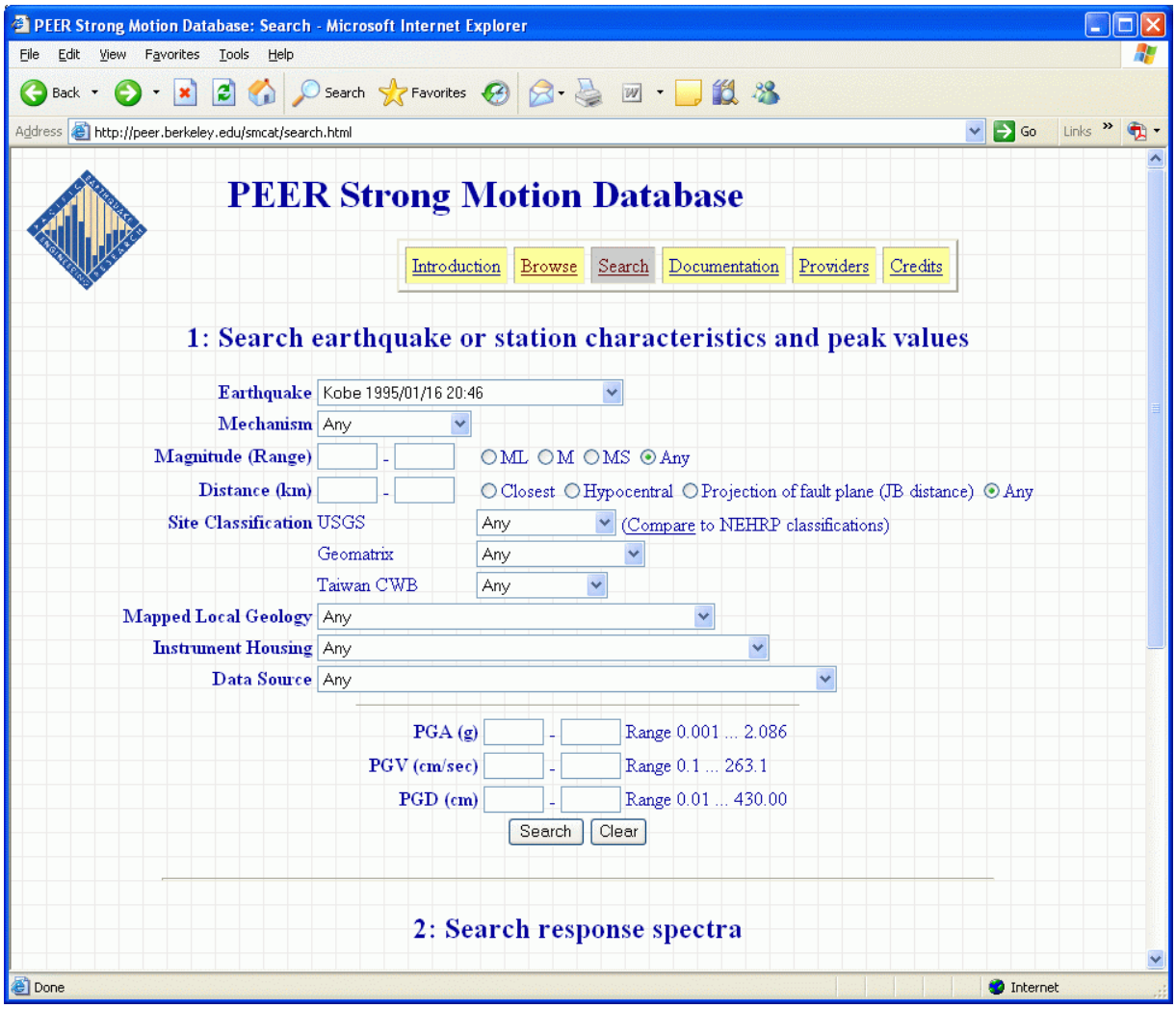
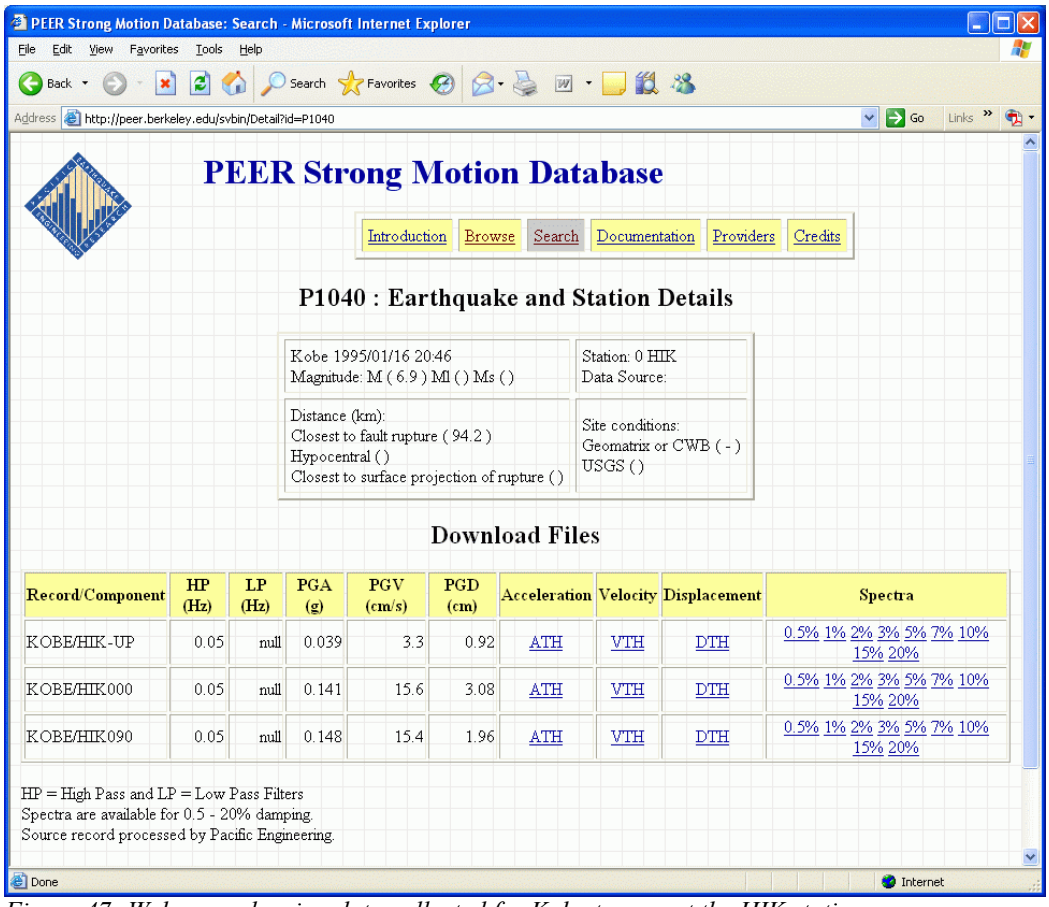
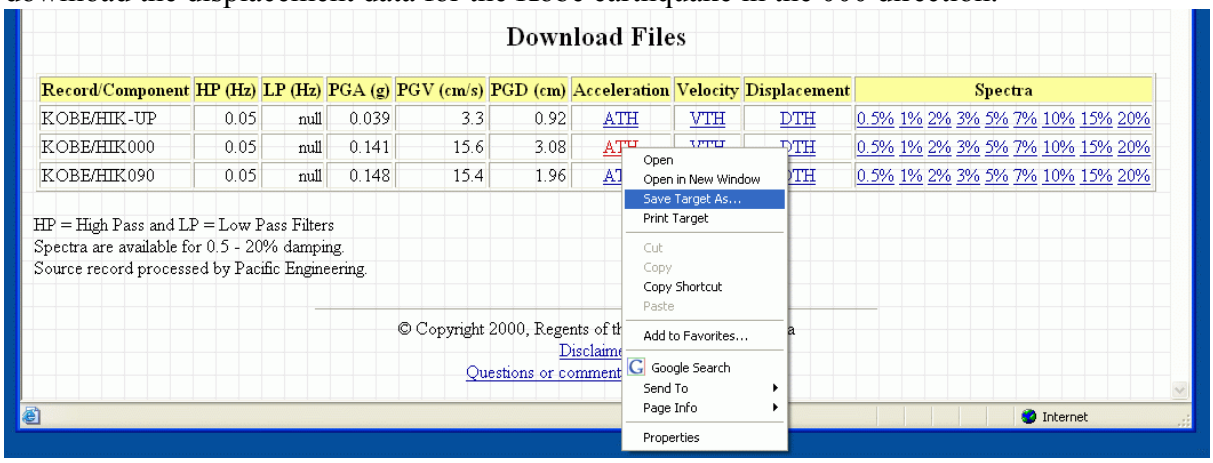

PEER Strong Motion Database: Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://peer.berkeley.edu/smcat/search.html
Go Links
PEER Strong Motion Database
Introduction Browse Search Documentation Providers Credits
1: Search earthquake or station characteristics and peak values
Earthquake Kobe 1995/01/16 20:46
Mechanism Any
Magnitude (Range) ML M MS Any
Distance (km) Closest Hypocentral Projection of fault plane (JB distance) Any
Site Classification USGS Any (Compare to NEHRP classifications)
Geomatrix Any
Taiwan CWB Any
Mapped Local Geology Any
Instrument Housing Any
Data Source Any
PGA (g) Range 0.001 ... 2.086
PGV (cm/sec) Range 0.1 ... 263.1
PGD (cm) Range 0.01 ... 430.00
Search Clear
2: Search response spectra
Done
Internet

Figure 46: Searching for an earthquake on the Berkeley website.

3. The *Query Results* page is displayed and it lists all the stations that recorded the Kobe earthquake. Click on *Record ID* P1040 to view the results from the HIK station. The *P1040: Earthquake and Station Details* web page shown in Figure 47 loads.


PEER Strong Motion Database: Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://peer.berkeley.edu/svbin/Detail?id=P1040
Go Links
PEER Strong Motion Database
Introduction Browse Search Documentation Providers Credits
P1040 : Earthquake and Station Details
Kobe 1995/01/16 20:46
Magnitude: M (6.9) Ml ( ) Ms ( )
Station: 0 HIK
Data Source:
Distance (km):
Closest to fault rupture (94.2)
Hypocentral ( )
Closest to surface projection of rupture ( )
Site conditions:
Geomatrix or CWB (-)
USGS ( )
Download Files
Record/Component HP (Hz) LP (Hz) PGA (g) PGV (cm/s) PGD (cm) Acceleration Velocity Displacement Spectra
KOBE/HIK-UP 0.05 null 0.039 3.3 0.92 ATH VTH DTH 0.5% 1% 2% 3% 5% 7% 10% 15% 20%
KOBE/HIK000 0.05 null 0.141 15.6 3.08 ATH VTH DTH 0.5% 1% 2% 3% 5% 7% 10% 15% 20%
KOBE/HIK090 0.05 null 0.148 15.4 1.96 ATH VTH DTH 0.5% 1% 2% 3% 5% 7% 10% 15% 20%
HP = High Pass and LP = Low Pass Filters
Spectra are available for 0.5 - 20% damping.
Source record processed by Pacific Engineering.
Done
Internet
Figure 47: Web page showing data collected for Kobe tremor at the HIK station.

4. As depicted in Figure 48, right-click on the *ATH* link under the KOBE/HIK000 record to download the displacement data for the Kobe earthquake in the 000 direction.


Download Files
Record/Component HP (Hz) LP (Hz) PGA (g) PGV (cm/s) PGD (cm) Acceleration Velocity Displacement Spectra
KOBE/HIK-UP 0.05 null 0.039 3.3 0.92 ATH VTH DTH 0.5% 1% 2% 3% 5% 7% 10% 15% 20%
KOBE/HIK000 0.05 null 0.141 15.6 3.08 ATH VTH DTH 0.5% 1% 2% 3% 5% 7% 10% 15% 20%
KOBE/HIK090 0.05 null 0.148 15.4 1.96 ATH VTH DTH 0.5% 1% 2% 3% 5% 7% 10% 15% 20%
HP = High Pass and LP = Low Pass Filters
Spectra are available for 0.5 - 20% damping.
Source record processed by Pacific Engineering.
Open
Open in New Window
Save Target As...
Print Target
Cut
Copy
Copy Shortcut
Paste
Add to Favorites...
Google Search
Send To
Page Info
Properties
Internet
Figure 48: Saving earthquake acceleration record.

5. Once the earthquake record is downloaded, save the AT2 text file in the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).

Whether through browsing or searching, the HIK000.AT2 file should now be downloaded. When opened in Matlab Editor, the HIK000.AT2 file appears as shown in Figure 49.

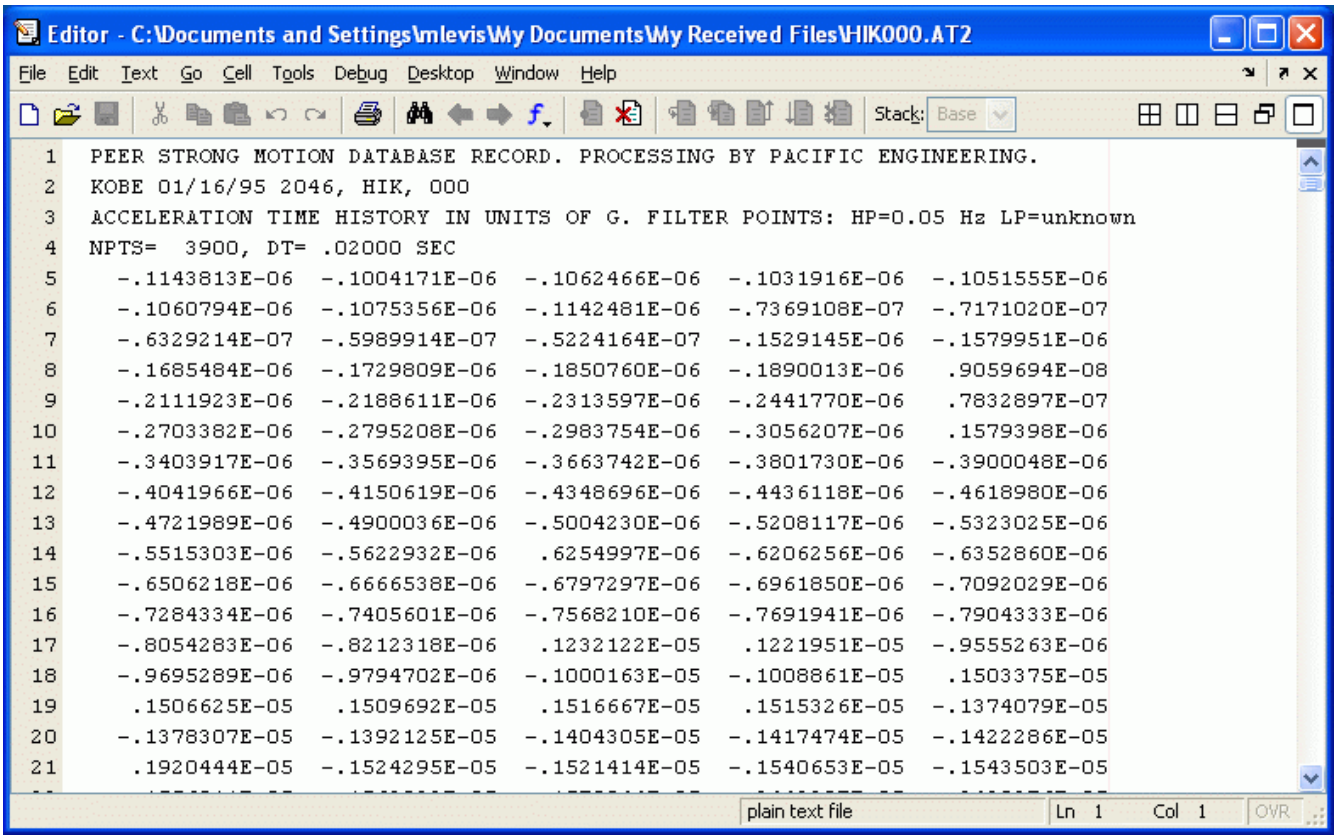


Figure 49: Raw earthquake data file HIK000.AT2 shown when opened in Matlab Editor.

6.4.4. FFT Evaluation: fft_eval_pos.m and fft_eval_acc.m

The Matlab script *fft_eval_pos.m* calculates the Fast Fourier Transform and generates a Bode plot of the desired and measured position data. Similarly, the *fft_eval_acc.m* script generates the Bode plot for the desired and measured acceleration data.

Follow these steps obtain a Bode plot of the position and acceleration data after the table has tracked a predefined trajectory:

1. Go through the procedure in Section 6.7.2 to run either a compound sine wave or an earthquake. Make sure that both the *x (m)* and *a_tbl (g)* scopes are open and the time buffer is long enough to capture the entire signal.
2. Once the experiment has been ran, save the collected position data from the *x (m)* scope into the MAT file *data_x.mat* and the acceleration data from the *a_tbl (g)* scope into *data_a.mat*. Data from a scope can be saved into an MAT file by choosing on the *File | Save | Save As MAT-File...* item under the scope menu bar (see .
3. Open the *fft_eval_pos.m* script in Matlab Editor. The “INPUT” section of the script is pictured in Text 11. Set the variable *fname* to the MAT file where the position data is going to be saved,

in this case *data_x.mat*. Also, adjust the minimum frequency, *f_min*, and maximum frequency, *f_max*, variables to obtain a Bode plot with the desired frequency range.

```
% ************************************************************************
% INPUT
% ************************************************************************
% Save collected positions in the following file.
fname = 'data_x.mat';
% min/max frequencies for plotting (Hz)
f_min = .1;
f_max = 20;
```

Text 11: "INPUT" section in fft_eval_x.m script.

4. As shown in the "LOAD DATA" in Text 12, the data is loaded into the Matlab workspace using the Matlab *load* command. The *power_spectrum.m* Matlab M-File function computes the power spectrum of a signal. The power spectrum of a signal is the absolute value of the signal FFT, e.g. for the frequency ω the power spectrum of *g(t)* is denoted $|G(\omega)|$.

```
% ************************************************************************
% LOAD DATA
% ************************************************************************
% Load desired and measured position into Matlab workspace.
load(fname);
% desired position (m);
x_des = q_data_q8_Scopes__Positions_x__m__0_;
% resulting measured table position (m)
x_meas = q_data_q8_Scopes__Positions_x__m__1_;
%
% ************************************************************************
% FFT
% ************************************************************************
% Sampling frequency (Hz)
f_s = 1 / default_sample_time;
% Calculate power spectrum of desired position
[fp,Pxd] = power_spectrum(x_des,f_s,f_min,f_max);
% Calculate power spectrum of measured position
[fp,Pxm] = power_spectrum(x_meas,f_s,f_min,f_max);
```

Text 12: "LOAD DATA" and "FFT" section in fft_eval_pos script.

5. Run the script to generate a Bode plot of the desired and collected position data. The plot shown in Figure 50 was generated with a predefined sine wave when *make_sine* was configured with frequency vector *fd* = [1, 4, 10]. As illustrated, the spikes of the power spectrum occur at these frequencies. The dashed blue trajectory is the desired or commanded position generated by the *make_sine* or *make_quake* scripts, and the solid red trajectory is the position measured by the encoder.

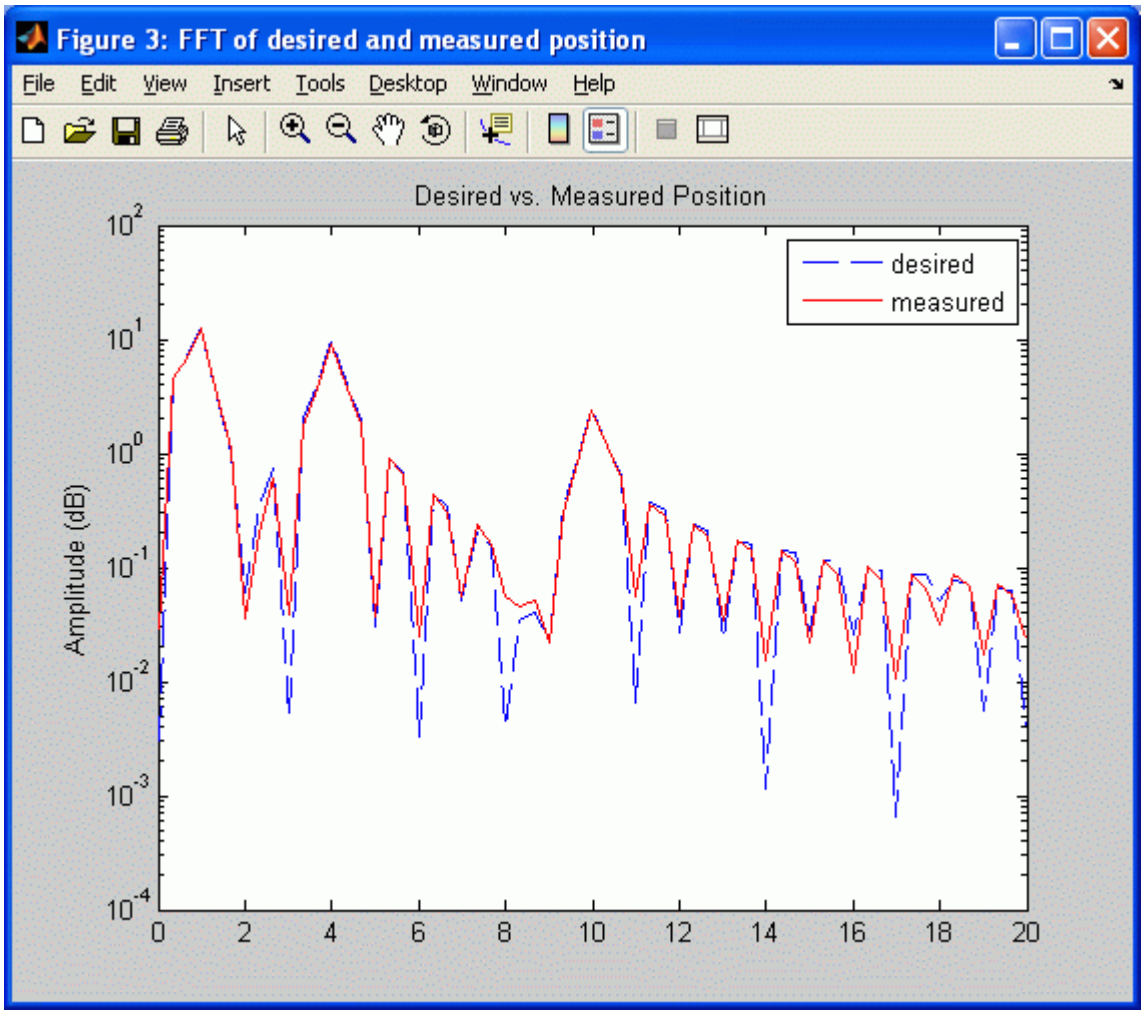


Figure 50: Bode plot of desired and measured stage position after running q_data_q8 with a predefined sine wave.

6. To obtain a Bode plot of the acceleration data, configure the *fft_eval_acc.m* script as with explained explained earlier in Step 3 for *fft_eval_x.m* file. Make sure the file name is set to *data_a.mat* for example and that the frequency range is set as desired.
7. Run the *fft_eval_acc.m* script. The bode plot shown in Figure 51 was generated after running the Northridge earthquake. The desired acceleration of the earthquake, which is the dash blue trace, is the recorded acceleration of the actual Northridge tremor (i.e. the acceleration from the earthquake data file). As illustrated, the accelerations measured on the Shake Table II accelerometer matches the recorded data quite well for frequencies under 10 Hz.

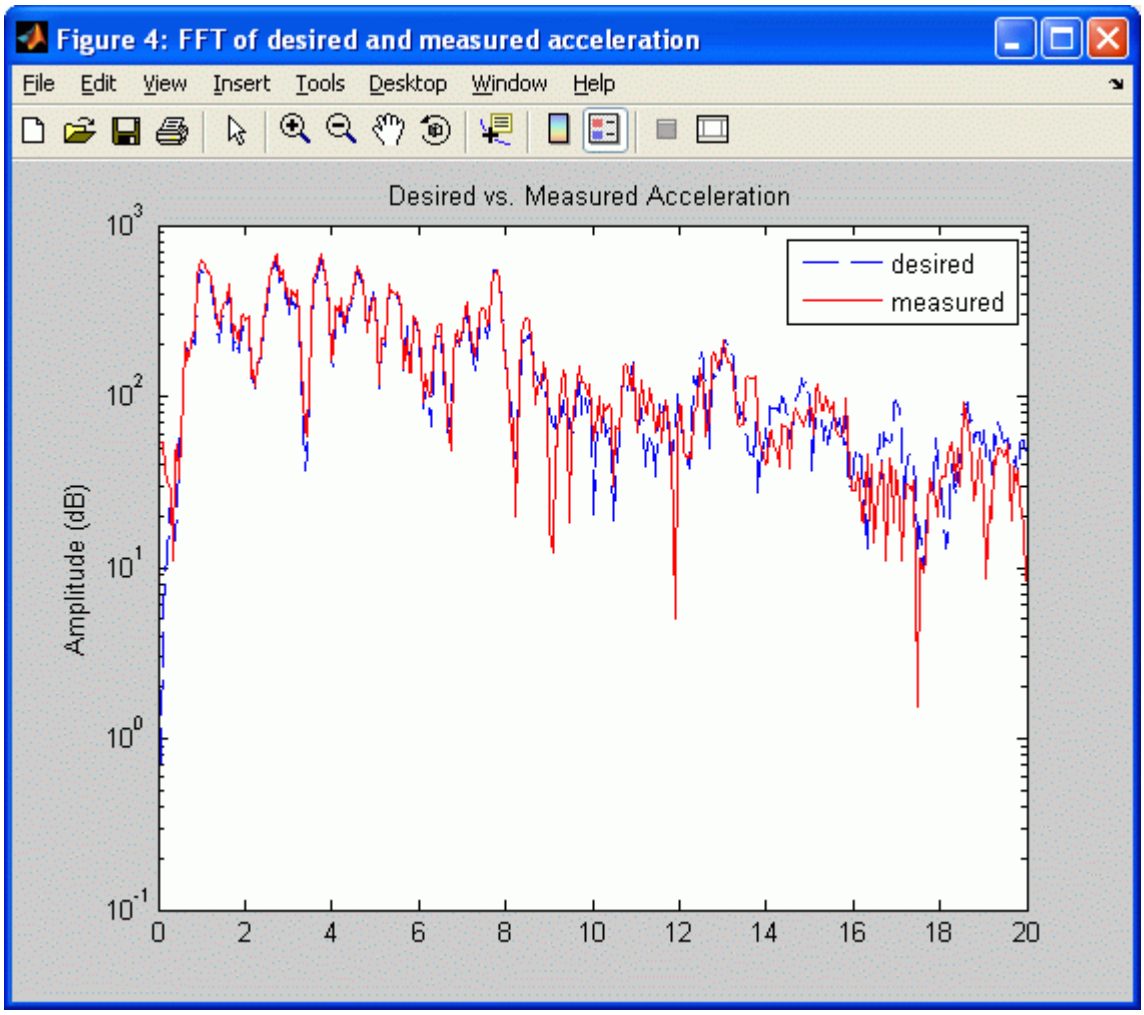


Figure 51: Bode plot of the Northridge earthquake acceleration data.

6.5. Sine Wave: q_sine_zz.mdl

The q_sine_zz.mdl Simulink diagram is used with WinCon to make the stage of the Shake Table II track a sine wave. The WinCon Project that is used in Section 5.2.3 was, for example, constructed using this file. Section 6.5.1 describes the different components of the Simulink diagram and Section 6.5.2 explains how to build and run a WinCon Controller from the file.

6.5.1. Model Description

The *q_sine_q8* Simulink diagram is depicted in Figure 52 below. The *Accelerometers – Q8*, *Enable UPM-180-25B - Q8*, *Limit Switches – Q8, Scopes: Positions*, and *Scopes: Accelerations* blocks are linked to the WinCon Shake Table II library and are explained in sections 6.3.2, 6.3.3, 6.3.4, and 6.3.8, respectively.

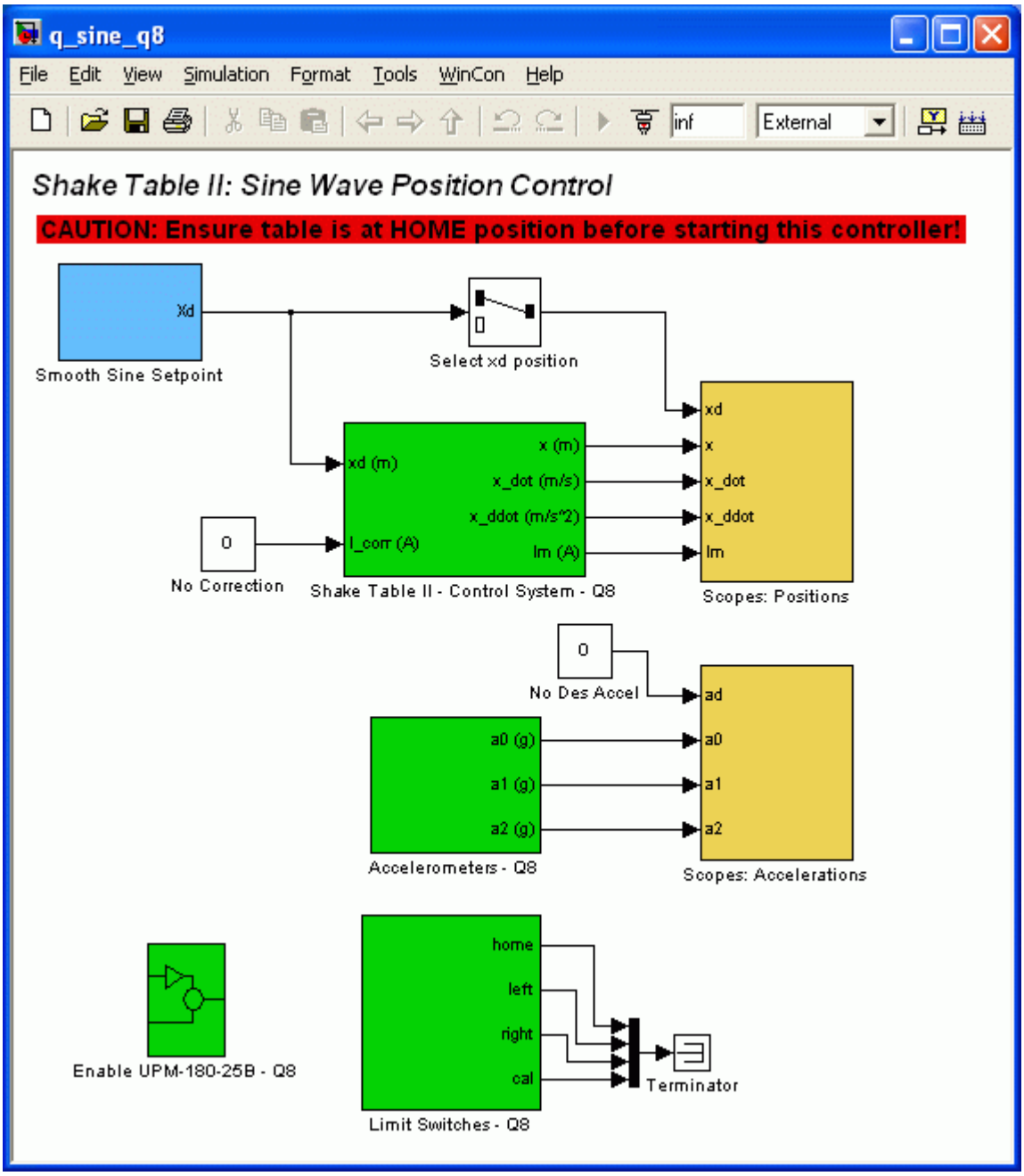


Figure 52: The q_sine_q8 Simulink Model used to control the stage position according to a user-specified sine wave..

6.5.1.1. Shake Table II – Control System Subsystem

The *Shake Table II – Control System – Q8* block is common to all the supplied Simulink models for the Quanser Q8 board and its interior is pictured in Figure 53. The *PD Position Controller* block, explained in Section 6.3.6, implements the position controller that calculates the necessary motor input current needed to move the ST II stage to the desired position. This current is fed to the Shake Table II – Q8

block which drives the motor through the UPM-180-25B amplifier and outputs the linear stage position from the encoder measurement. See Section 6.3.1 for an description of the Shake Table II – Q8 block. As shown in Figure 52 there is no correction current, *I_corr (A)*, for this model. Correction current is only used in q_data_zz model when the command position is predefined and is explained in Section 6.7. This subsystem outputs the resulting measured current *Im (A)*, the stage position measurement *x (m)*, the stage velocity measurement *x_dot (m/s)*, and the acceleration of the stage calculated using the encoder *x_ddot (m/s^2)*. See Section 6.3.7 for information on the *Calculate Acceleration from Encoder* subsystem.

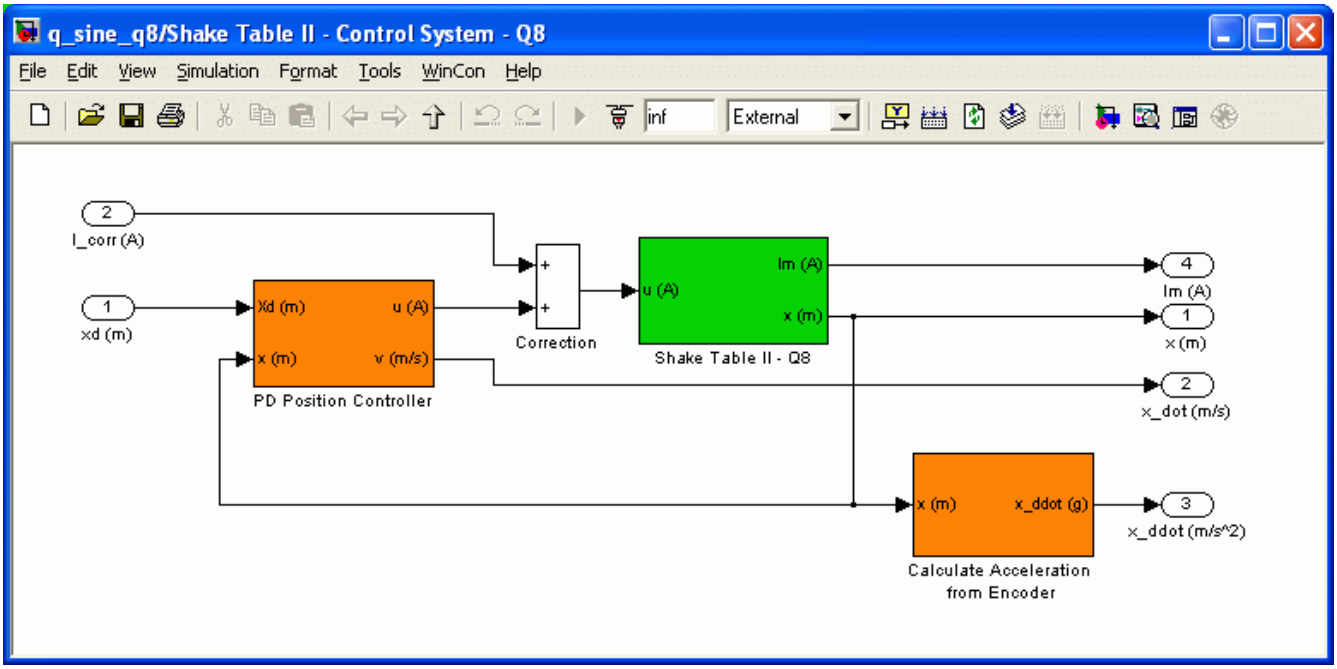


Figure 53: Shake Table II - Control System - Q8 subsystem.

6.5.1.2. Smooth Sine Setpoint Subsystem

The *Smooth Sine Setpoint* block, shown in Figure 54, is used to generate a sine wave position setpoint that is smooth and constrained to the Shake Table II limits. The actual sine wave is generated using the Smooth Sine Wave Simulink block found in the Quanser Toolbox. It gradually changes the sine wave when the amplitude or frequency of the signal is changed. This eliminates any “jumps” in the setpoint and prevents any large current spikes to enter the ST II motor.

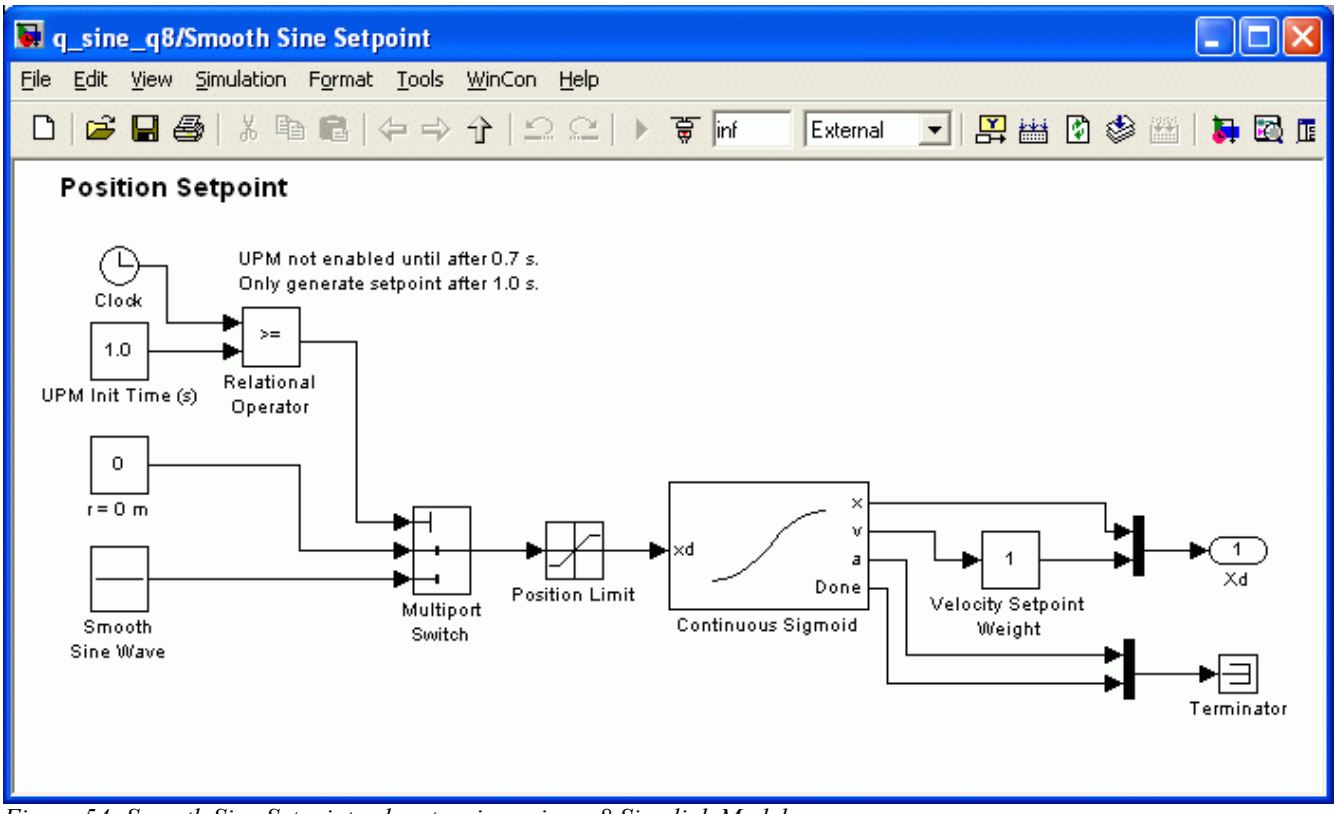


Figure 54: Smooth Sine Setpoint subsystem in q_sine_q8 Simulink Model.

The power amplifier in the UPM-180-25B device is only enabled after 0.7 seconds of starting the WinCon Controller (see the *Enable* block description in Section 6.3.4). As a result, the setpoint is set to 0 for the first second regardless of the *Smooth Sine Wave* settings. This is done using the Relational Operator and Multiport Switch blocks shown in Figure 54.

The Saturation block labeled *Position Limit* constrains the setpoint to the table stroke limits P_MAX. The Continuous Sigmoid block is found in the Quanser Toolbox under the *Sources* category and it limits the velocity and acceleration of its input signal to user-specifications. In this case, the sine wave command position is constrained to the Shake Table II limits VEL_MAX and ACC_MAX, which are set by the *stii_limits* script. The sigmoid makes the sine wave smoother (helps prevent large spikes in the controller output) and it also computes the desired stage velocity setpoint. The *Velocity Set-Point Weight* slider block determines how much (if any) of the velocity setpoint to use. It is the b_{sd} parameter in the PD controller shown in Section 6.3.6.

6.5.2. Building and Running Sine Wave WinCon Controller

Follow these steps to build and run the q_sine_zz WinCon Controller:

1. Make sure the amplifier has been initialized as discussed in Section 5.2.1.
2. **CAUTION: Ensure table is at *HOME* position before running this experiment!** Otherwise the experiment may stop prematurely because the table reached the left or right limit sensors. If

the *Home* LED on the front panel of the blue UPM-180-25B is **not** lit, then go through the calibration procedure in Section 5.2.2.

3. Load Matlab.
4. Through the *Current Directory* window, go to the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).
5. Run the *setup.m* file script. See Section 6.4.1 for more information on configuring and running this script.
6. Open the q_sine_zz.mdl file.
7. Click on the *WinCon* menu found in the q_sine_zz Simulink diagram menu bar,.
8. Select the *Build* item to begin generating the WinCon Controller. The message shown in Figure 55 will load indicating that the code from the model is being generated.

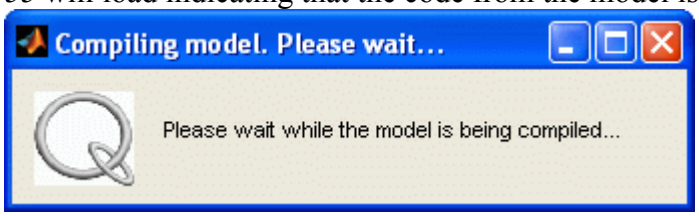


Figure 55: Compiling model message.

9. Once the q_sine_zz WinCon Controller has been generated, the WinCon Server window shown in Figure 56 will load.

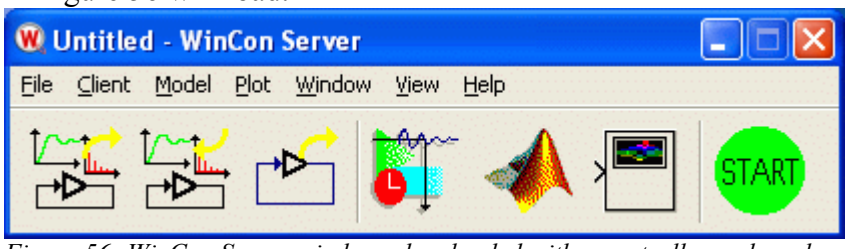


Figure 56: WinCon Server window when loaded with a controller and ready to run.

10. Click on the green START button in the WinCon Server to run the controller.
11. The amplitude of the sine wave is, by default, initially set to 0 meters. Open the *Smooth Sine Setpoint* subsystem, shown in Figure 54, and double-click on the Smooth Sine Wave source block.
12. In the dialog box, set the *Amplitude* to 10e-3 meters and the Frequency to 1 Hz. The stage should begin tracking a ±10 cm sine wave running at 1 Hz.

CAUTION: **The sine wave amplitude is in meters! Do not set the *Amplitude* parameter in the Smooth Sine Wave block to any value larger than 65e-3 meters, i.e. 65 cm.**

13. Before stopping the controller, set the Amplitude back to 0e-3 to place the stage back at home position.
14. Click on the red STOP button in the WinCon Server window to stop running the controller.
15. Close the WinCon Project by selecting *File* | *Close* in WinCon Server.
16. If no more experiments will be conducted in this session, turn off the UPM-180-25B power.

6.6. Sine Sweep: q_sweep_zz.mdl

The q_sweep_zz.mdl Simulink diagram is used with WinCon to make the stage of the Shake Table II track a sine sweep signal. The q_sweep_zz.wcp WinCon Project in Section 5.2.4 was constructed using this file. See Section 6.6.1 for a description of the Simulink diagram and Section 6.6.2 for the procedure on building and running its corresponding WinCon Controller.

6.6.1. Model Description

The q_sweep_zz.mdl Simulink model is pictured in Figure 57 below. The *Accelerometers – Q8*, *Enable UPM-180-25B - Q8*, *Limit Switches – Q8, Scopes: Positions*, and *Scopes: Accelerations* blocks are linked to the WinCon Shake Table II library and are explained in sections 6.3.2, 6.3.3, 6.3.4, and 6.3.8, respectively. The Shake Table II – Control System – Q8 subsystem implements the position control and was previously discussed in Section 6.5.1.1.

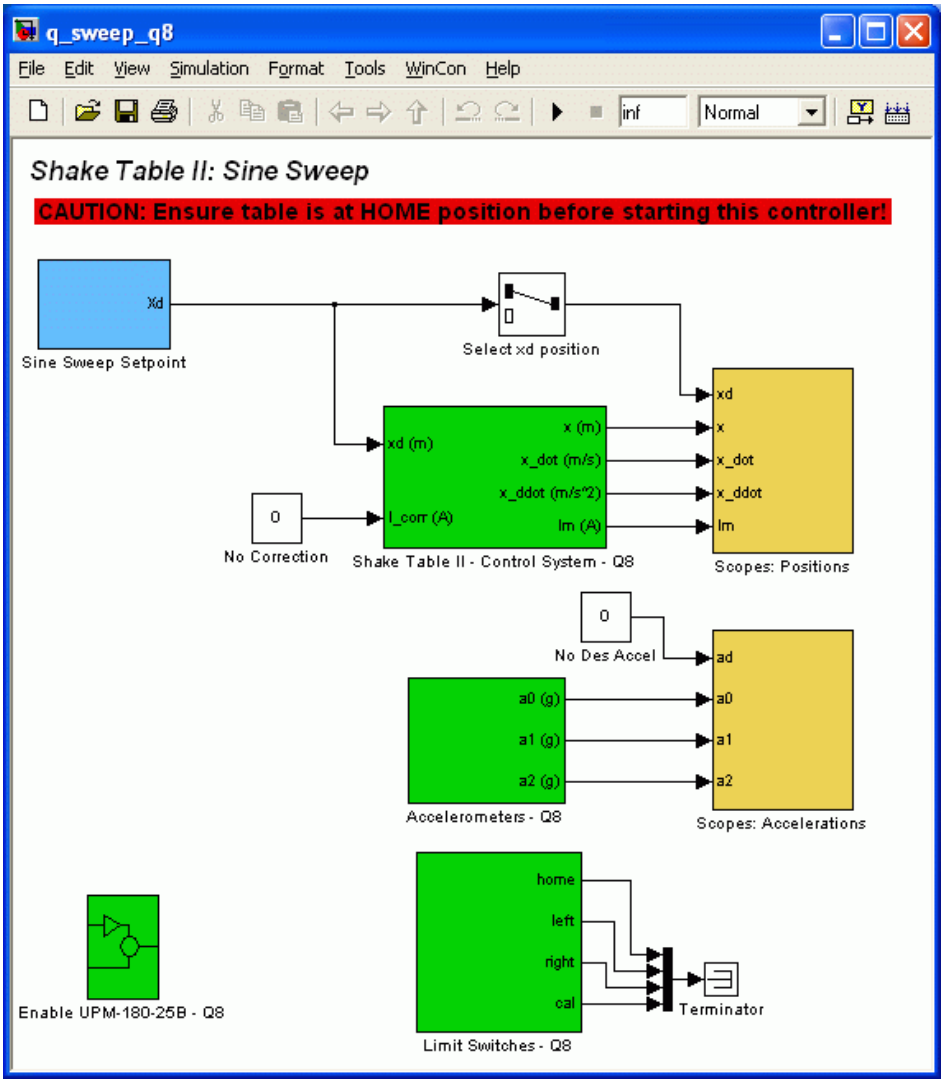


Figure 57: The q_sweep_q8 Simulink model used for sine sweep position control on the ST II.

The sine sweep signal is generated by the *Sine Sweep Setpoint* subsystem shown in Figure 58. More specifically, the sweep signal is generated by the Quanser Chirp block. This outputs a sine wave with a fixed amplitude that increases in frequency over time. The initial frequency, target time (duration of sweep), and the final frequency at the target time can all be specified in the block parameters.

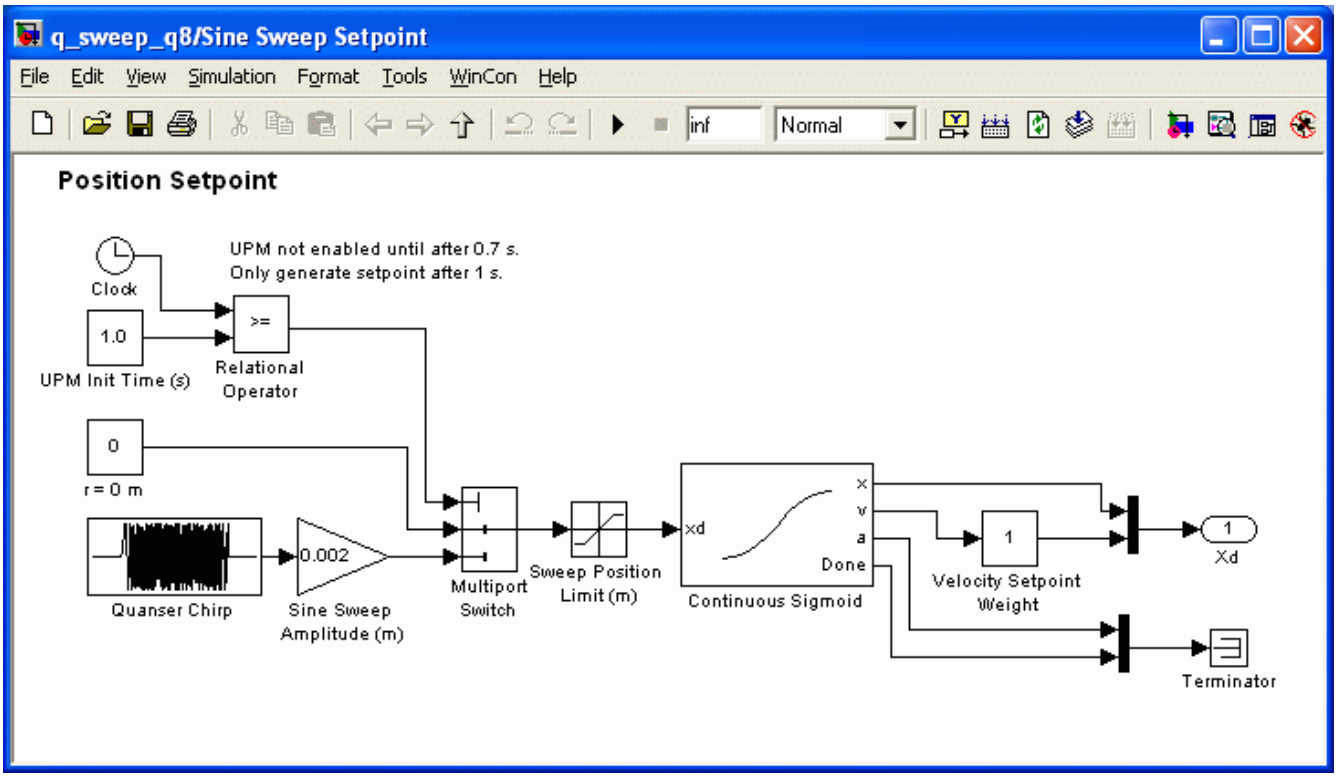


Figure 58: Sine Sweep Setpoint subsystem in q_sweep_q8 Simulink Model.

Similarly to the *Smooth Sine Setpoint* block used in the q_sine_zz Simulink diagram (described in Section 6.5), the setpoint command is only outputted after 1 second to allow enough time for the amplifier to be enabled. The position is limited by the *Sweep Position Limit (m)* saturation block with the SWEEP_MAX parameter that is defined in the *setup.m* script. By default, the SWEEP_MAX is set to 4 mm for safety purposes. The Continuous Sigmoid block makes the chirp signal smoother and limits its velocity and acceleration to the ST II limits. The *Velocity Setpoint Weight* slider gain adjusts the amount of velocity reference to be used by the PD control.

6.6.2. Building and Running Sine Sweep WinCon Controller

Go through the following procedure to build and run the q_sweep_zz WinCon Controller:

1. Make sure the amplifier has been initialized as discussed in Section 5.2.1.
2. **CAUTION: Ensure table is at *HOME* position before running this experiment!** Otherwise the experiment may stop prematurely because the table reached the left or right limit sensors. If the *Home* LED on the front panel of the blue UPM-180-25B is **not** lit, then go through the calibration procedure in Section 5.2.2.

3. Load Matlab.
4. Through the *Current Directory* window, go to the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).
5. Run the *setup.m* file script. See Section 6.4.1 for more information on configuring and running this script.
6. Open the q_sweep_zz.mdl Simulink diagram, shown in Figure 57 above.
7. From the Simulink Diagram menu bar, select the *WinCon* | *Build* item.
8. Once the *q_sweep_zz* WinCon Controller has been generated, the WinCon Server window will load.
9. As shown in Figure 58, the amplitude of the sine sweep is, by default, initially set to 0.002 meters. To change the sweep amplitude, go into the *Sine Sweep Setpoint* subsystem and change the *Sine Sweep Amplitude (m)* gain block.

WARNING: Do not set the *Sine Sweep Amplitude (m)* block to any value larger than 0.003 meters unless a low frequency is being used.

10. Click on the green START button in the WinCon Server to run the controller. The table should begin moving back-and-forth faster and faster.
11. The sweep will restart automatically. To stop running the controller, click on the red STOP button in the WinCon Server window.
12. [Optional] If desired, select *File* | *Save As* in the WinCon Server window to save it as a WinCon Project file.
13. When done, select *File* | *Close* in WinCon Server to close the controller (or *File* | *Save As* to save it as a project).
14. If no more experiments will be conducted in this session, turn off the UPM-180-25B power.

6.7. Predefined Data: q_data_zz.mdl

The q_data_zz Simulink Model is used to replay predefined trajectory data on the shake table. This is usually done after running either the make_quake or make_sine Matlab scripts. For example, the Northridge or Kobe WinCon Projects described in Section 5.2.5 were built using q_data_zz. Section 6.7.1 describes this Simulink diagram and the procedure to build and run its WinCon Controller is given in Section 6.7.2.

6.7.1. Model Description

The q_data_q8 Simulink model is shown in Figure 59. Most of these blocks are linked from the WinCon Shake Table II library and explained in Section 6.3. The *Accelerometers – Q8*, *Enable UPM-180-25B - Q8*, *Limit Switches – Q8, Scopes: Positions*, *Scopes: Accelerations*, and *Stop controller after duration* blocks are linked to the WinCon Shake Table II library and are explained in sections 6.3.2, 6.3.3, 6.3.4, 6.3.8, and 6.3.9, respectively.

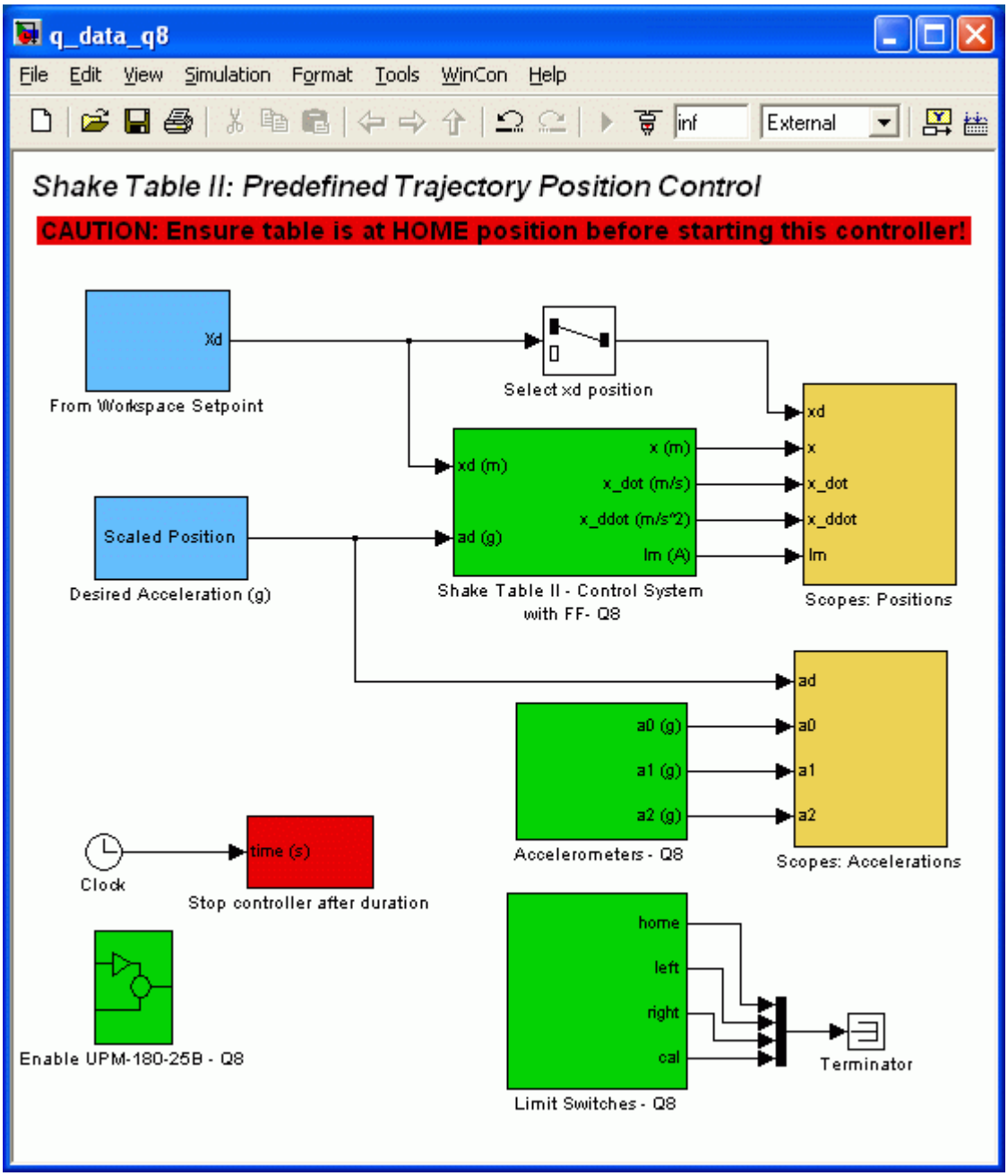


Figure 59: The q_data_q8 Simulink model used for predefined trajectory position control on the ST II.

The *From Workspace Setpoint* block outputs the setpoint position that is loaded in the Matlab workspace and its interior is shown in Figure 60. As in the setpoint subsystems used for the sine wave and sine sweep signals, explained in sections 6.5.1 and 6.6.1, the *From Workspace Setpoint* block only generates the setpoint 1 second after the controller has started and imposes the position, velocity, and acceleration limits using the saturation and Continuous Sigmoid blocks. Note also that velocity set-point weight is set to 1. The *Desired Position (m)* is a Repeating Sequence block found in the Simulink toolbox under the *Sources* library. It outputs the array [*t*, *xd*], where *t* is the time and *xd* is the desired

position, that is stored in the Matlab workspace. This data can be loaded using the make_sine.m or make_quake.m scripts.

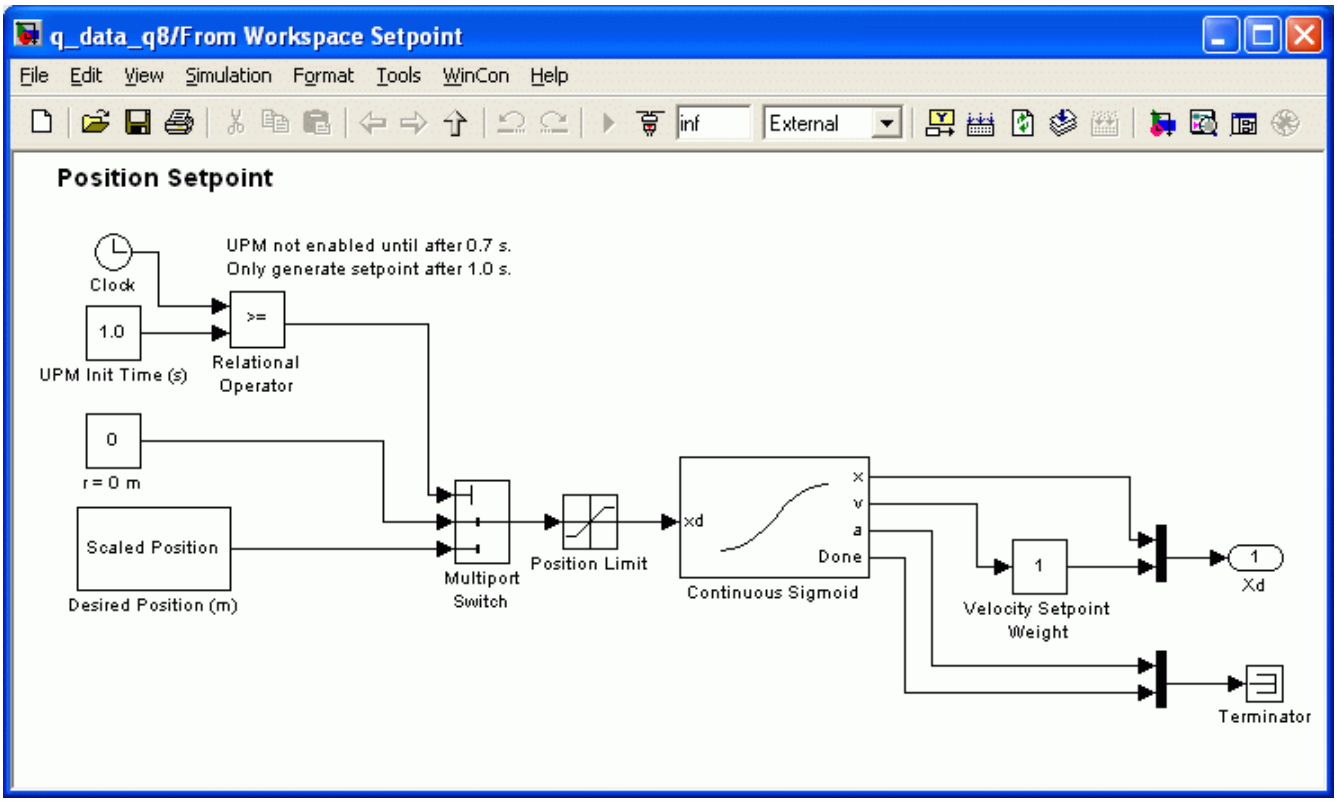


Figure 60: From Workspace Setpoint block in q_data_zz Simulink Diagram.

The *Desired Acceleration (g)* Repeating Sequence block shown in Figure 59 outputs the array [*t*, *ad*], where *t* is the time of the trajectory and *ad* is the desired acceleration. When the make_sine and make_quake scripts are ran, they load both the desired position, *xd*, and the desired acceleration, *ad*, as explained in sections 6.4.2 and 6.4.3, respectively.

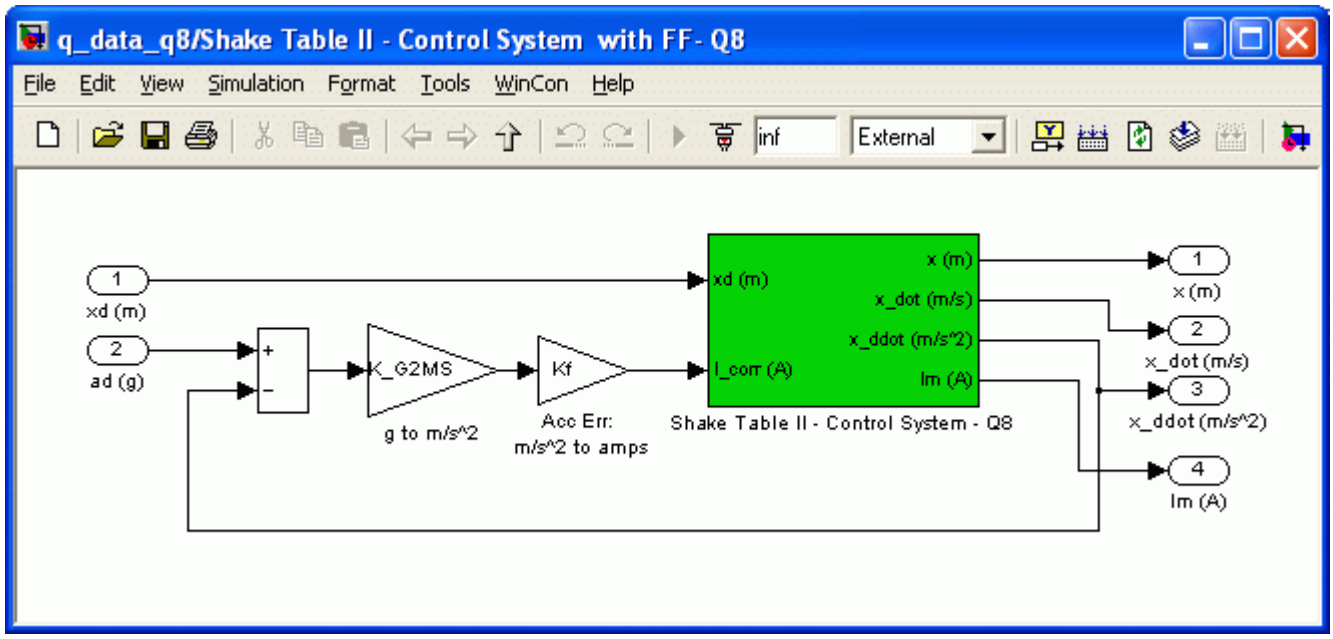


Figure 61: Shake Table II - Control System with FF - Q8 subsystem in q_data_q8 Simulink Diagram.

The desired acceleration is used in the *Shake Table II – Control System with FF* block, depicted in Figure 61, to attain better position control performance. The *Shake Table II – Control System – Q8* subsystem implements the proportional-derivative (PD) position controller and is discussed in Section 6.5.1.1. The *FF* in the subsystem label stands for feed-forward and as shown in Figure 62 it accompanies the PD controller. Feed-forward takes advantage of the fact that the setpoint or command position is predefined to help minimize lag and make the position tracking more responsive. Given the currently measured acceleration and the desired acceleration, the feed-forward gain computes the current needed to move the stage in order to attain the desired acceleration. Remark, as depicted in Figure 61, that the acceleration calculated from the encoder is used for acceleration feedback and not the accelerometer measurement.

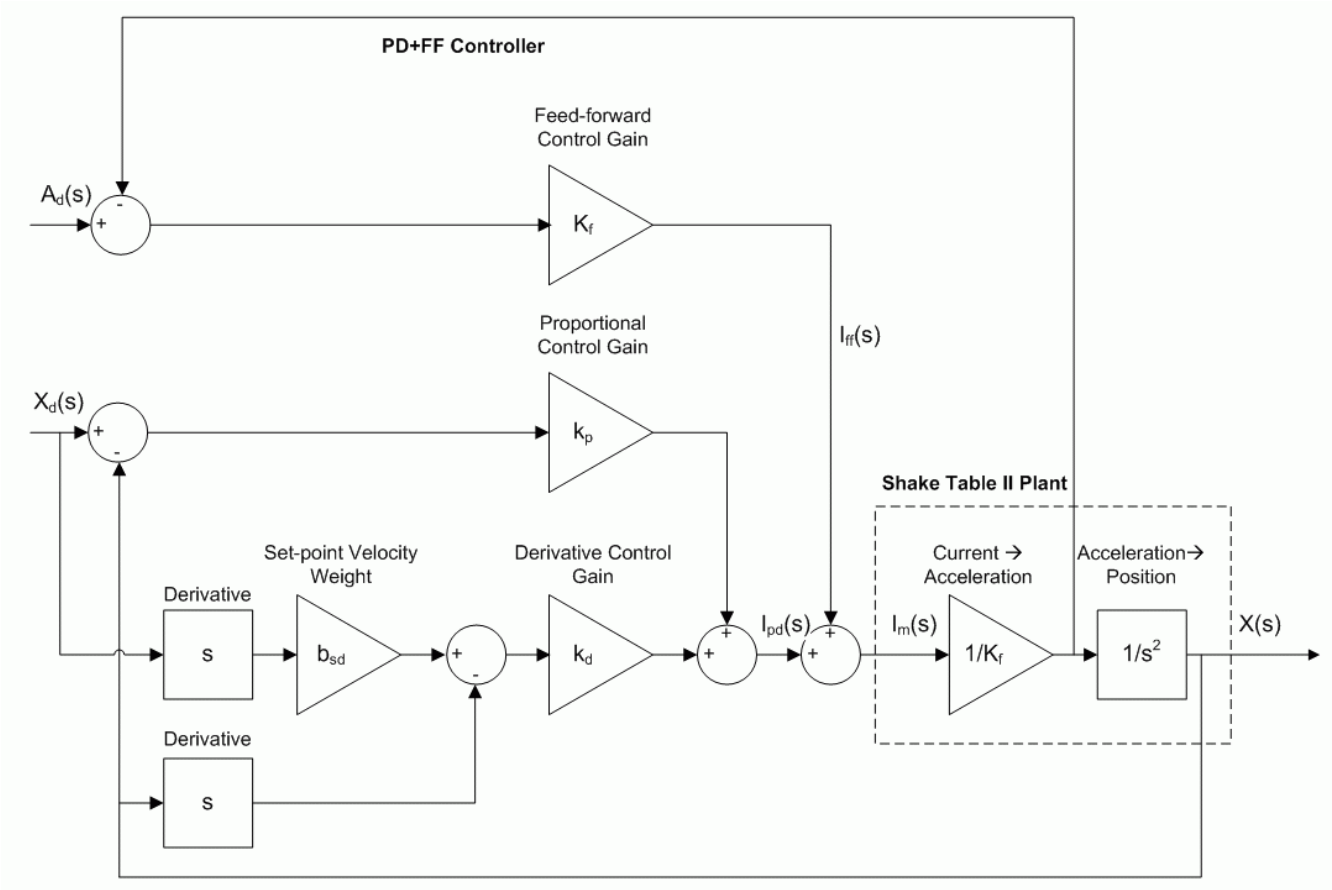


Figure 62: Block diagram of proportional-derivative plus feed-forward position ST II control system.

6.7.2. Building and Running Predefined Trajectory WinCon Controller

Follow these steps to build and run the predefined trajectory position controller:

1. Make sure the amplifier has been initialized as discussed in Section 5.2.1.

2. **CAUTION: Ensure table is at *HOME* position before running this experiment!** Otherwise the experiment may stop prematurely because the table reached the left or right limit sensors. If the *Home* LED on the front panel of the blue UPM-180-25B is **not** lit, then go through the calibration procedure in Section 5.2.2.
3. Load Matlab.
4. Through the *Current Directory* window, go to the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).
5. Run the *setup.m* file script. See Section 6.4.1 for more information on configuring and running this script.
6. Run either the *make_sine.m* or *make_quake.m* scripts. See Section 6.4.2 for details on generating a composite sine wave with the *make_sine* script and Section 6.4.3 for more information on replaying an earthquake on the shake table with *make_quake*.
7. Open the *q_data_zz.mdl* Simulink diagram. The q_data_q8 model is shown above in Figure 59.

8. From the Simulink Diagram menu bar, select the *WinCon | Build* item.
9. Once the controller has been generated, the WinCon Server window will load.
10. In the WinCon Server window, click on the *Open Scope* button and select the *x (m)* under the *Scopes: Positions* folder and the *a_tbl (g)* variables under the *Scopes: Accelerations* folder. The *x (m)* scope displays the desired and measured position of the stage. The *a_tbl (g)* scope displays the desired acceleration and the acceleration of the stage measured by the accelerometer.
11. Make sure the time buffer of both scopes are equal to or greater than the duration time of the sine wave or earthquake trajectory. To check the duration of the time buffer, enter *tf* in the Matlab prompt. The time buffer of a scope can be changed by clicking on the *Update* menu and selecting the *Buffer...* item. The *Select Buffer Size* window should load. See Section 5.2.6 for more information.
12. Start the controller by clicking on the green *START* button in the WinCon Server window. The table should begin tracking the trajectory that was generated by the *make_sine* or *make_quake* scripts.
13. Figure 63 shows a typical desktop setup after running a *q_data_q8* controller with a predefined sine wave, i.e. using *make_sine*. The *q_data_zz* controller stops automatically when the duration of the sine wave, earthquake, or any other predefined trajectory is reached. The bode plot of the position or acceleration data can be generated using the *fft_eval_pos* or *fft_eval_acc* scripts, as explained in Section 6.4.4.

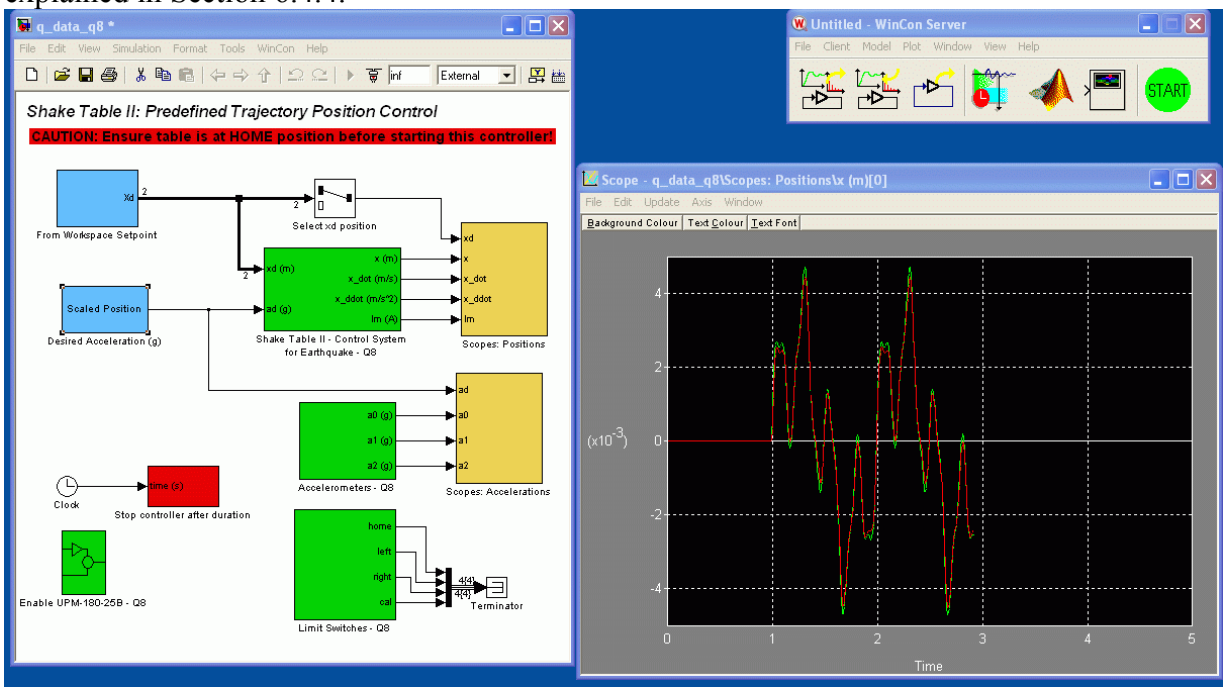


Figure 63: Desktop view after running a sine wave with the q_data_q8 controller.

15. [Optional] If desired, select *File | Save As* in the WinCon Server window to save it as a WinCon Project file. See Section 6.7.3 for instructions on how to make a WCP file.
16. When done, select *File | Close* in WinCon Server to close the controller (or *File | Save As* to save it as a project).

17. If no more experiments will be conducted in this session, turn off the UPM-180-25B power.

6.7.3. Constructing a WinCon Project

In order to compile different WinCon projects, such as q_kobe_zz.wcp and q_northridge_zz.wcp, the q_data_zz Simulink diagram needs to be saved under a different name. A WinCon Controller is associated with a single Simulink Model. Thus when going through the procedure outlined in Section 6.7.2 the WinCon Controller called q_data_zz.wcl is generated no matter if the Kobe or Northridge data is loaded and what the name of the WinCon Project is.

Follow this procedure to create different WinCon Projects from the q_data_zz.mdl Simulink model

1. Load Matlab.
2. Through the *Current Directory* window, go to the *STII\Lab Files\zz\mdl* folder (where *zz* is q4 or q8) on your PC (which was copied from the ST II CD).
3. Open the q_data_zz.mdl Simulink model.
4. In the Simulink Diagram menu bar, select *File* | *Save As* and save *q_data_zz.mdl* as another file describing the earthquake. For example, *q_elcen_zz.mdl* for the El Centro tremor.
5. Go through steps 5-17 in the procedure described in Section 6.7.2 for the new Simulink file above to build its associated WinCon Controller. Feel free to open new scopes (see Section 5.2.6) and/or design control panels using WinCon Controls. Go to Reference [3] for more information on designing control panels.
6. Select *File* | *Save As* in the *WinCon Server* window to save this package as a WinCon Project, or WCP, file. The opened plots and data loaded in the workspace along with the WinCon Controller are now all saved under the project, just like in the q_northridge.wcp and q_kobe.wcp projects.
7. Click on File | Close in WinCon Server to close the project.
8. To load the compiled project, load WinCon Server, select *File* | Open, and locate the WCP file. See the procedure to run a tremor in Section 5.2.5 for more details. Note that Matlab does not need to be opened since the code has already been generated.

7. References

[1] Quanser Inc. *Q4 or Q8 User Manual*.
[2] Quanser Inc. *Universal Power Module User Manual*.
[3] Quanser Inc. *WinCon User Manual*.
[4] Danaher Motion. *AKM Series Motors*. Page 19 in the AKM24F column of *AKM_Motor_Selection_Guide.pdf*.
[5] Quanser Inc. *AMD-1 User Manual*.
[6] Quanser Inc. *AMD-2 User Manual*.
[7] Quanser Inc. *Shake Table II with AMD-1 User Manual*.
[8] Quanser Inc. *Shake Table II with AMD-2 User Manual*.
[9] Quanser Inc. *Shake Table II with 2xAMD-1 User Manual*.
[10] Quanser Inc. *WinCon Installation Manual*.

Appendix A. Troubleshooting Guide

This section provides the user with a list of solutions to the questions that may occur when setting up the Shake Table II system.

Q1. Why are the *Left* and *Right* LEDs on the UPM flashing?

The UPM has not been initialized yet. See Section 5.2.1 for the UPM180-25B initialization procedure.

Q2. Why is the *Left* or *Right* LED lit after undergoing the UPM initialization procedure?

The stage of the table is at one of the ends of the table and has triggered the *Left* or *Right* proximity sensor. See the procedure in Section 5.2.2 to calibrate the stage to the *Home* position

Q3. Why is the *OK* LED on the UPM not lit when running a WinCon project?

The E-Stop button is either pressed down or improperly connected to the UPM. Stop the WinCon controller and power off the UPM. Then, verify that the E-stop button is in the released upright position (i.e. wind the knob clockwise indicated by the arrows) and confirm that it is properly connected to the UPM. See Section 4.2.3 for more information on connecting the E-Stop to the UPM-180-25B device.

Q4. Why is the shake table not moving when running *q_cal_zz.wcp, q_sine_zz.wcp, q_sweep_z.wcp, q_kobe_zz.wcp*, or *q_northridge_zz.wcp* WinCon projects?

- **Is the red power LED in the top-left corner of the UPM lit?**

  If not turn the switch to the OFF position and ensure the AC cord is securely connected. If after switching the UPM ON the LED is still not lit, the fuse may be blown. Replace the fuse and try re-powering the UPM.
- **Is the red LED on the terminal board lit?**

  If not then the fuse may be blown, there may be a lack of power being supplied to the terminal board, or some other problem is associated with the board. See the corresponding data acquisition card manual for details on handling this situation.
- Go through the calibration procedure detailed in Section 5.2.2. If the table does not move towards the *Home* position and the OK LED on the UPM is off then the see **Q3** solution and then try to calibrate the shake table again.
- If the table still does not move to the *Home* position when calibrating, go through Section 4.2.3 and verify all the connections. In particular, verify the wiring from the blue UPM-180-25B device to the shake table.

Appendix B. Q8 Extended Terminal Board Signals

The Quanser Q8 Extended Terminal Board, pictured in Figure 10, does *not* have separate analog input channels and does not have the Encoder Input 0 and Encoder Input 4 connectors. The A/D channels 0-3 and Encoder Input Channel 0 are integrated in the *Table X* connection and the A/D channels 4-7, and Encoder Input Channel 4 are integrated in the *Table Y* connection. The limit detector signals – *Left* , *Home*, and *Right* – and the *Calibrate* and *Enable* signals from the PIC in the UPM are also carried in the *Table X* connection. Table 15 specifies the various signals carried between the UPM and *Table X* connection on the Q8 Extended Terminal Board.

Signal	*WinCon Interface*	*Description*
D/A #0	Analog Output: Channel 0	Drives the amplifier in the *x*-axis UPM.
A/D #0	Analog Input: Channel 0	Analog sensor (i.e. accelerometer) connected to *S1* on *x*-axis UPM.
A/D #1	Analog Input: Channel 1	Analog sensor connected to *S2* on *x*-axis UPM.
A/D #2	Analog Input: Channel 2	Analog sensor connected to *S3* on *x*-axis UPM.
A/D #3	Analog Input: Channel 3	Analog sensor connected to *S4* on *x*-axis UPM.
Encoder #0	Encoder Input: Channel 4	Measurement from ST II motor encoder.
DI #0	Digital Input: Channel 0	*Left* limit detector signal.
DI #1	Digital Input: Channel 1	*Home* limit detector signal.
DI #2	Digital Input: Channel 2	*Right* limit detector signal.
DI #3	Digital Input: Channel 3	*Calibrate* signal.
DO #0	Digital Output: Channel 8	Sends *Calibrate* signal to PIC on *x*-axis UPM.
DO #1	Digital Output: Channel 9	Sends *Enable* signal to PIC on *x*-axis UPM.

Table 15: Signals in the Table X connection.

A second *y*-axis shake table can be coupled with an *x*-axis shake table. In this configuration, a second UPM is required for the *y*-axis table and the signals between this UPM and the *Table Y* connector on the terminal board are listed in Table 16.

Signal	*WinCon Interface*	*Description*
D/A #4	Analog Output: Channel 4	Drives the amplifier in the UPM.
A/D #4	Analog Input: Channel 4	Analog sensor (i.e. accelerometer) connected to *S1* on *y*-axis UPM.
A/D #5	Analog Input: Channel 5	Analog sensor connected to *S2* on *y*-axis UPM.

Signal	*WinCon Interface*	*Description*
A/D #6	Analog Input: Channel 6	Analog sensor connected to *S3* on *y*-axis UPM.
A/D #7	Analog Input: Channel 7	Analog sensor connected to *S4* on *y*-axis UPM.
Encoder #4	Encoder Input: Channel 4	Measurement from the ST II motor encoder.
DI #4	Digital Input: Channel 4	*Left* limit detector signal.
DI #5	Digital Input: Channel 5	*Home* limit detector signal.
DI #6	Digital Input: Channel 6	*Right* limit detector signal.
DI #7	Digital Input: Channel 7	*Calibrate* signal.
DO #10	Digital Output: Channel 10	Sends *Calibrate* signal to PIC on *y*-axis UPM.
DO #11	Digital Output: Channel 11	Sends *Enable* signal to PIC on *y*-axis UPM.

Table 16: Signals in Table Y connection.